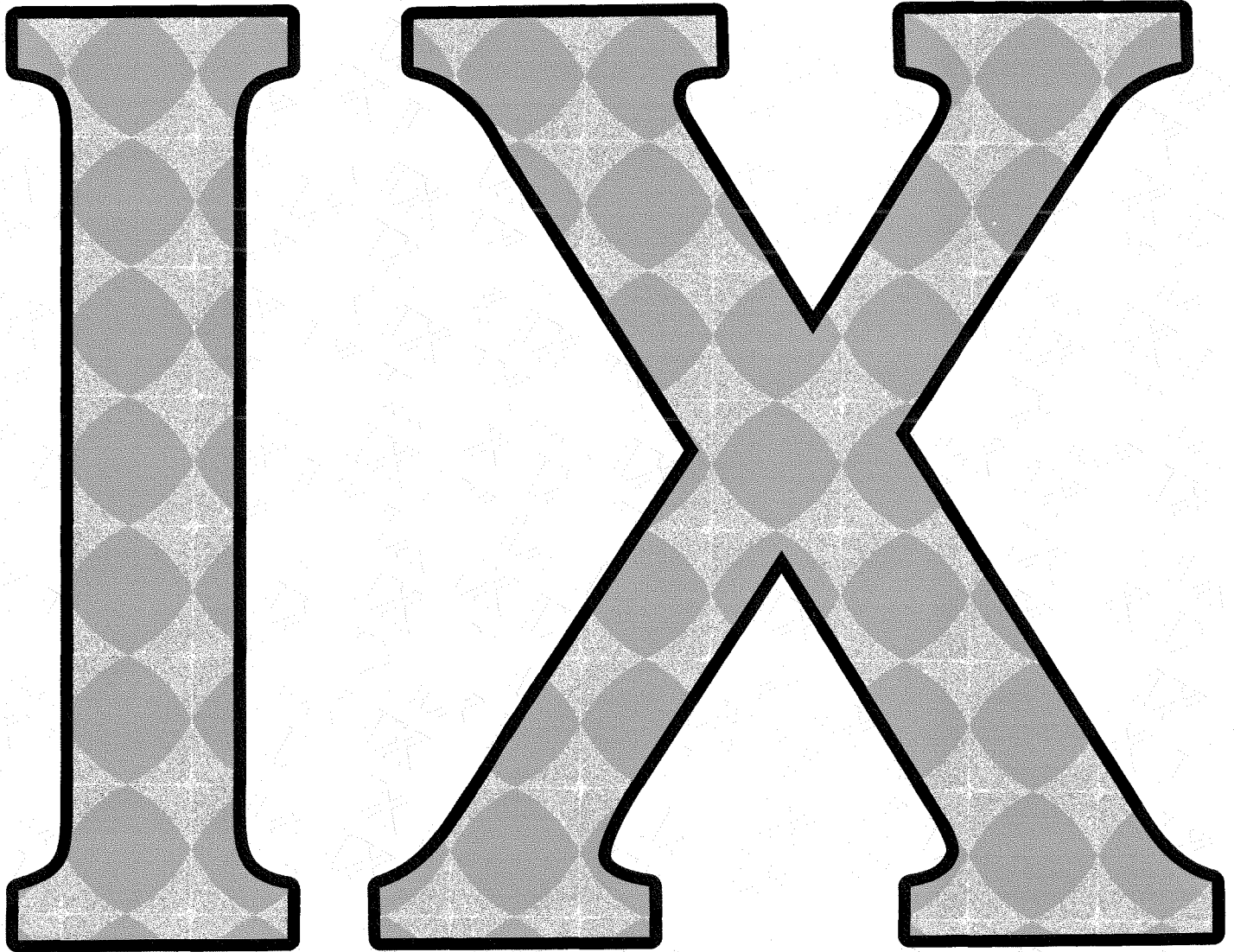


QL Today

Volume 9
Issue 1
May/June
2004

ISSN 1432-5454

The Magazine about QL, QDOS,
Sinclair Computers, SMSQ...



Nearly Tradition: the first issue of a Volume starts with the Volume number. And No: we have not turned into a UNIX magazine!

Thanks to you, dear trusty Reader, it looks very much as if we can celebrate the round "10" (or "X") in a year's time. But first e the brandnew first Issue of the new Volur

Contents

- 3 Editorial
- 4 News
- 8 QDT US Edition *Marcel Kilgus*
- 9 Gee Graphics! (on the QL ?) - part 38
H. L. Schaaf
- 11 The Epson "LQ 300+" Printer
Paul L. Harris
- 12 Bad Boys Rape or We PIN-DOWN?
Al Boehm
- 13 QL show - Orlando, Florida, USA.
Tony Firshman
- 15 What Now? *David Denham*
- 18 Programming in Assembler: Part 10 -
Linked Lists *Norman Dunbar*
- 31 Using a CD-ROM on the Q40
Timothy Swenson
- 32 Quanta Workshop and AGM 2004
Dilwyn Jones
- 36 What are blobs and patterns?
Wolfgang Lenerz
- 38 Letter Box
- 42 Some Solutions to the Dreaded Printing
Problem *John Gregory*
- 43 Caption Competition *Dilwyn Jones*
- 44 Disassembling Register Lists
George Gwilt
- 47 Machine Code Tally Sort
Stephen Poole
- 50 Byts of Wood *Roy Wood*
- 55 QL 2004

Advertisers

in alphabetical order

Dilwyn Jones	7
Jochen Merz Software	33
QBranch	26, 27
Quanta	21
RWAP	45
TF Services	41
Geoff Wicks	11

QL Today

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software Tel. +49 203 502011
Im stillen Winkel 12 Fax +49 203 502012
47169 Duisburg BBS +49 203 502014
Germany email: JMerz@j-m-s.com
 email: QLToday@j-m-s.com

English office:

Q Branch Tel. +44 1273 386030
20 Locks Hill Mobile +44 7836 745501
Portslade Fax +44 1273 381577
BN41 2LB email: qbranch@qbranch.demon.co.uk
United Kingdom email: QLToday@j-m-s.com

Editor:

Dilwyn Jones email: dilwyn.jones@tesco.net
41 Bro Emrys email: QLToday@J-M-S.com
Tal-Y-Bont, Bangor
Gwynedd
United Kingdom LL57 3YT

Co-Editor:

Bruce Nicholls Tel +44 1708 510764
38 Derham Gardens Fax +44 870 0568755
Upminster email: qltoday@q-v-d.demon.co.uk
Essex RM14 3HA email: QLToday@j-m-s.com
United Kingdom

QL Today is published bi-monthly, our volume begins at the beginning of June. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email or into one of the JMS-BBS's. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising deadlines are as follows:

- Issue 1: 30 April
- Issue 2: 30 June
- Issue 3: 30 August
- Issue 4: 30 October
- Issue 5: 30 December
- Issue 6: 28 February

QL Today reserves the right to publish or not publish any material submitted. Under no circumstances will **QL Today** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL Today**. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

This magazine and all material within is © copyright 2003 Jochen Merz Software unless otherwise stated. Written permission is required from the publisher before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

Welcome to the ninth volume of QL Today. Little did I think nine years ago that I'd still be here at the helm after all this time, but I am and enjoying myself! And after all this time, the QL scene still manages to throw up some pleasant surprises. In this issue we have a review of a program which is, I think, unique to the QL. I don't think I've ever seen a program like PIN-Down, which generates a rhyming sequence of words to help you remember PIN numbers. In Britain, we are in the process of converting credit card transactions into Chip and PIN formats, so there will be a lot of PIN numbers to remember. Just the program at just the right time.

Jim Hunkins's QDT desktop for QL systems with GD2 and SMSQ/E was demonstrated at the USA QL show and those who saw it speak highly of it. I have read enough about it to know it will be worth the wait and will change the way we use our QL systems. While I did not get to attend the USA show, I did go to Quanta's AGM in Manchester and bring you pictures with this issue, and Tony Firshman reports from the USA QL show.

Later this year, there will be a major event called QL2004 in The Netherlands. Held at the well known Sint Joris college in Eindhoven, this will be the biggest QL show of the year, hopefully attended by QLers from all over Europe and possibly further afield as well! It's to celebrate 20 years of QLing or 21 years of Quanta (Quanta was set up a little bit before most of us could get our hands on a QL). Geoff Wicks and others have been very active in preparing for this event and he has set up a website with advance information, much like Jim Hunkins did for the USA QL show.

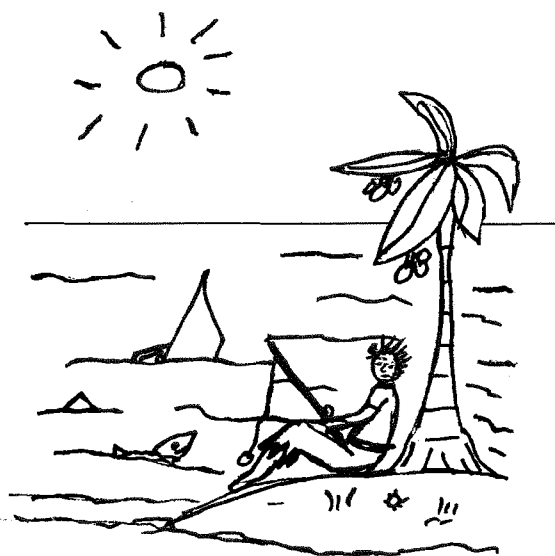
Every month seems to bring more and more QL related web sites

and we do our best to inform you in the news pages of QL Today. With progress still being made with soql and other internet access systems (uqlx and qemulator have varying degrees of support for such systems, while Peter Graf continues working on his system) it is to be hoped that once the basic systems are up and running and available, that applications will soon follow.

Don't forget our caption competition in this issue...turn to page 43 to win yourself some software! Poor Jim Hunkins, as if the pressure of writing QDT wasn't enough, he had to feature in a caption competition as well.

Thank you all our loyal readership for your support over the years and of course thank you to our authors who keep coming up with articles for each issue. Special thanks to Herb Schaaf and Norman Dunbar for their long running series of course - Herb alone has now clocked up 38 graphics articles! Keep up the good work everyone. Here's to another happy year of QLing.

Just my luck...all the time in the world and not a QL in sight!!!!



Cartoon

NEWS

Quanta News

Quanta's Annual General Meeting took place at the Manchester Workshop on April 18th 2004. The new officers and committee members were returned as follows:

Chairman - John Mason

Secretary - John Southern

Treasurer - John Gilpin

Committee members - Roy Brereton, John Gregory and Geoff Wicks.

Roy Brereton is currently acting as editor of the newsletter following the resignation of Paul Merdian. John Gregory remains as Software Editor, while John Gilpin remains as Head Librarian.

Acting Editor Roy Brereton has appealed for members' contributions to the newsletter - contributions can be sent to him by email - editor@quanta.org.uk or on disk by post at 94 Teignmouth Road, Clevedon, North Somerset, BS21 6DR, England.

PIN-Down

Geoff Wicks (see picture) of Just Words! has released a new program called PIN-Down.

As the name suggests, this is an aide-memoire to help you generate random rhyming phrases to remember those PIN (Personal Identity Numbers) so prevalent in modern life, anything from credit or debit card numbers to burglar alarms and even computers.

PIN-Down generates phrases for PIN-Numbers up to 10 digits long and I can say that it is a really easy to use program. It is pointer driven and in the usual Just Words! house style. Best of all, it costs just five pounds.

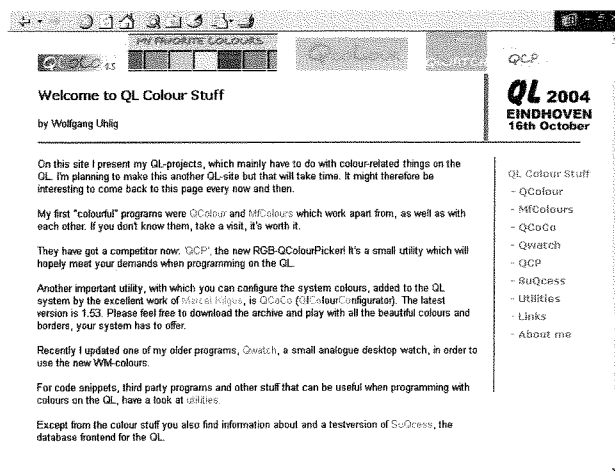
Wolfgang Uhlig's QL Website



Wolfgang Uhlig (see picture) has released a number of utilities for GD2 ("colour driver") systems, including the QCoCo, QColour and Mfcolours ("My Favourite Colours") and has set up a web site from where you can get information

about and download these programs. It also includes a small program called Q-Watch, a small analogue clock program for your desktop. The site also includes a page about the SuQcess database program.

<http://www.uhlig.nl/ql/>



Wolfgang Uhlig's website

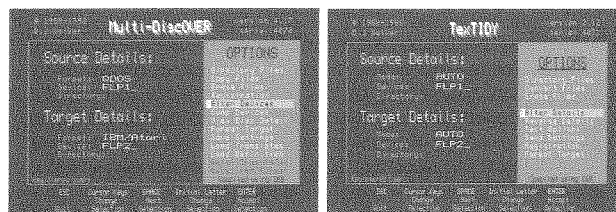
Discover and Textidy

Rich Mellor writes:

I have been in touch with Dave Walker (author of Discover and Textidy) about these two programs, and he has now released both programs as freeware. Although you need to obtain a registration key to make full use of the programs, there is no longer any fee to obtain this.

Programs can be downloaded from:

<http://homepage.ntlworld.com/itimpi/djwdown.htm>



Discover

Textidy

Aurora News



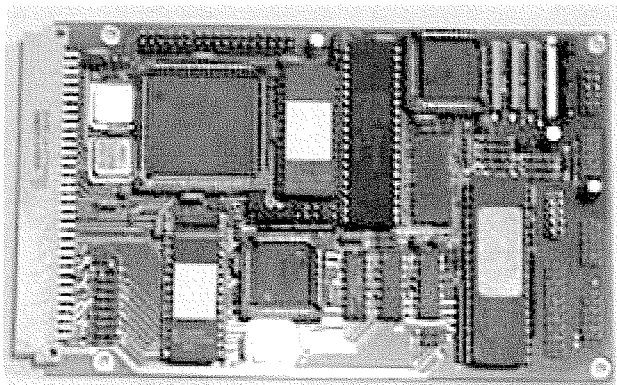
Zeljko Nastasic (Nasta, see picture) recently wrote on the QL-Users mailing list:

"There is one small project going on at the moment. There is a number of half-built Auroras that I have here (about 25), which will be completed, and sold off (hopefully) at very reasonable prices, mainly as a fund-riser for new projects." "These last Auroras will also be a sort of 'limited special editions', slightly upgraded with respect to the original Aurora. I should be finishing the expe-

rimental phase on this by the end of the month. These boards will come with the full shrouded IDC connectors (so no possibility of plugging stuff the wrong way), including the 'keyboard' connector, turned pin sockets for the ROM and IPC, and, a half megabyte Flash ROM in the ROM socket."

"The idea was to make it possible to boot directly into SMSQ/E but the way the Flash is interfaced is such that one could have any boot operating system in it, initially the boards will probably come with Minerva, now that it is PD. The Flash ROM is of course in-system programmable for software updates or upgrades, but it needs to be initially programmed on a programmer - eventually, it should be possible to make a 'boot loader' of sorts that would allow programming of the Flash with the OS(s) of choice, this is all down to software support (though not a lot of it)."

"There will also be an option to activate 32 additional lines of pixels in the 16 and 256 color mode, if the last 16k of the IO area on SGC can be freed up (usually this is where you would put Qubide if the ROM slot is to be used, but since SuperIDE uses a mechanism where no IO addresses are used at all, it is likely to be free) - this requires a minor change in the color drivers. The maximum resolutions would therefore be 1024x512 in 16 colors, and a rather odd 512x512 in 256 colors. Not extremely usefull since we do not have a 16 color driver (yet?) but the capability was actually there from the start. The boards also come already modified for GoldFire (regular Auroras need removal of one component, which is trivial, it's not fitted on these specials)."



The original Aurora board

SMSQ/E V3.06

Wolfgang Lernerz (the SMSQ/E registrar) writes: SMSQE 3.06 is ready - by the time you read this, you'll be able to get it from the usual resellers.

The sources are at

<http://www.scp-paulet-lernerz.com/smsqe/>

QL 2004

With plans for the QL2004 event at Sint Joris College in Eindhoven, The Netherlands, now well advanced, Geoff Wicks has set up a website with full details of this major event, including pictures of the venue. The event itself, probably the QL "Main Event" of 2004, is expected to be held on 16th October, with the possibility of another major event in the UK in 2005.

The QL2004 website is now available at:

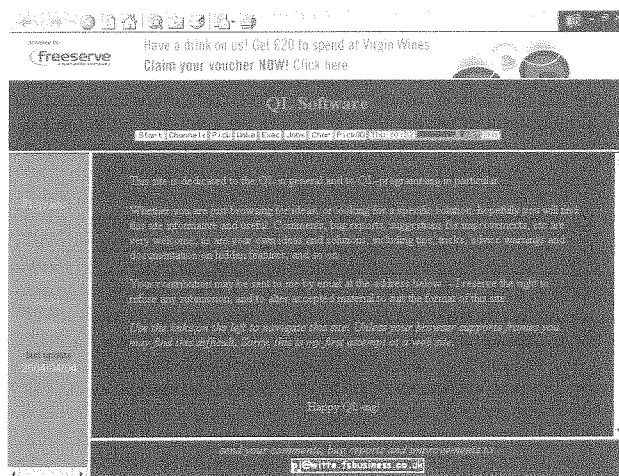
<http://members.lycos.co.uk/geoffwicks/ql2004.htm>

Per Witte's Website



Per Witte (see picture) has set up a website for his QL software, using the name Knoware. The site includes several downloadable QL programs, including some file and disk utilities, sprites and menu viewers and games, including some specifically GD2 programs. Visit his website at

<http://knoware.mysite.freeserve.com>



Per's website

QWIRC

The Qwirc (QL Winchester Rename Console) software by Per Witte has recently been updated. V0.64 works again in QL-colour mode 4, reduced memory consumption and some minor cosmetic changes. Version 0.63 removed JOB_ID not required, and fixed problems with unsightly borders in quickselect menus.

Version 0.62 has mainly cosmetic changes, while version 0.60 updated the program to use the new WMAN and GD2 facilities, and V0.61 tightened up

error trapping, doesn't crash if a drive is in use during startup, statuses displayed are read from drive not program variables, added Feedback button to About, fixed display error after window move and added Per's version of popup help (HIT and Hold).

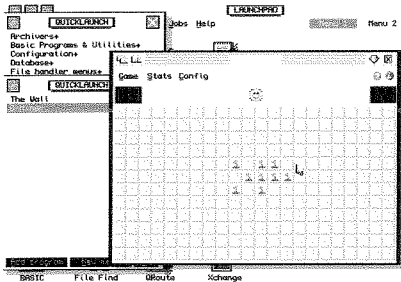
win	Name	Total (KB)	Free
win1	WIN1	511.99	295.68
win2	< free >		
win3	WIN3	100.00	80.01
win4	WIN4	95.00	18.07
win5	WIN5	300.00	153.33
win6	WIN6	450.00	33.84
win7	WIN7	599.99	168.48
win8	WIN8	50.00	18.05

The program will be available from Per's website and most sources of good QL free software.

Main screen from Qwirc

D-MINER

D-Miner v0.09 will soon be available from Per Witte. This is a Minesweeper type game, where you have to neutralise all hidden mines without getting blown up first. What makes this different to other Minesweeper games for the QL is that this one makes full use of GD2 and the new window manager, so it's colourful and detailed. It's so new I haven't really had much chance to play



with it (this news arrived just in time for this issue) so here's a screen dump to whet your appetite!

Screen dump from D-Miner

Spanish Printers Page



Javier Guerra (see picture) has made available a Spanish translation of my QL Printers web page. It may be found on the Sinclair QL Spanish Resources pages at

<http://sinclairql.info/docs/hardware/impresoras.htm>

WIKIPEDIA

Laurence Reeves wrote:

"I've just (well, nearly just) found Wikipedia and I couldn't resist correcting the QDOS link on http://en.wikipedia.org/wiki/Sinclair_QL to not go to <http://en.wikipedia.org/wiki/QDOS>, as that's the ancient Intel precursor of MS DOS."

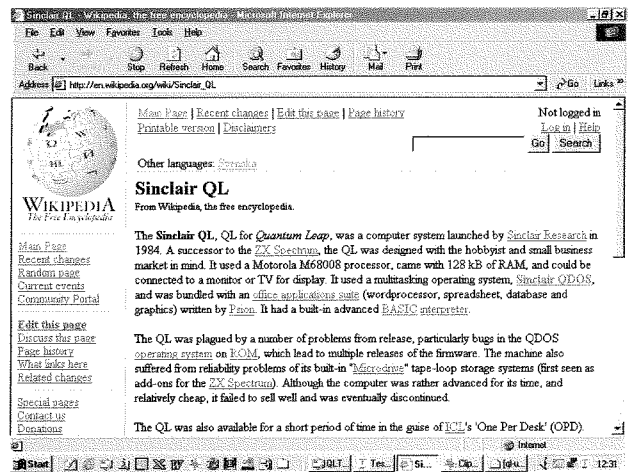
"I guess we could all have a go at improving the QL entry - adding lots of links back, or even start really using the Wiki."

"I couldn't even bring myself to add a single link, out of lack of confidence."

Wikipedia is a free online encyclopedia. It was started in January 2001 and about a quarter of a million pages are currently online or being prepared. It is possible for readers to contribute material and even edit pages - there is also a Swedish version of the QL page. As Laurence implies above, it may be possible for us to add lots of QL material and links if we all set our minds on it. The more QL material out there, the better.



The Wikipedia website (above) and QL page (below)



Launchpad Sprites

Dilwyn Jones writes:

I hope to make available the set of sprites I used in preparing Launchpad so that anyone wishing to write software consistent in appearance with the Launchpad environment can freely use my graphics. The sprite files (mostly 4-colour standard pointer environment sprites) will be available from my website, PD library or by email on request to anyone who wishes to use them.

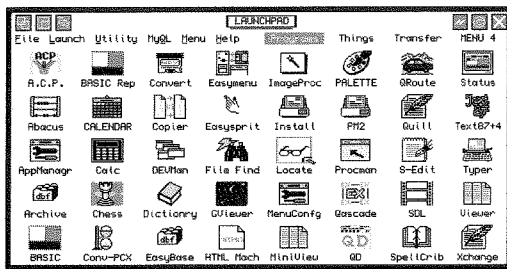
Dave Walker's Website

Please note that Dave Walker's website (source of QL software such as C68 compiler, Discover and Textidy) has moved, it is now at:

<http://homepage.ntlworld.com/itimpi/index.htm>

LAUNCHPAD

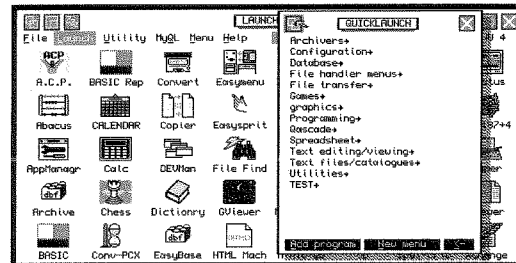
A New Graphical Interface For The QL



An all-new graphical user interface for QDOS or SMSQ/E from Dilwyn Jones. Fed up of typing in EXEC 'filename'? With this pointer driven application, simply set up desktop icons to launch programs, or create menus or "lists" of programs to launch, all in a simple to use no-nonsense graphical "point and click" system – use with mouse or keyboard. Runs on anything from expanded memory QL to QPC2 or Q60, as long as you have pointer environment and Toolkit 2

- Set up icons for launching your programs with a single mouse click from one of four desktop surfaces
- Set up your own program launching menu
- Up to 16 users, all with optional passwords
- MyQL menu to customise and remember your QL settings for each user
- Runs as a simple job, does not prevent you using BASIC or anything else
- Several accessory programs (calculator, calendar, screen saver, file handler, games and so on)
- Revolutionise your QL system – you'll wonder how you managed without it!

£20.00



THE QL CD-ROMS ARE STILL AVAILABLE...

QL Emulators CD £5.00 - QL Documentation CD £5.00 - QL PD-CDR £5.00
QL PD Library CD £10.00 - QL Religion CD £10.00 - QL Literature CD £10.00
Line Design Clipart CD £10.00 - Famous Faces Clipart CD for Line Design £10.00
PD Software Library catalogue available on my website <http://homepages.tesco.net/dilwyn.jones/index.html>

Want To Know More?

Visit the Launchpad web page at:
<http://homepages.tesco.net/dilwyn.jones/launchpad/launchpad.html>
from where you can download a free trial version, limited only by the number of programs you can set up to run on it – more than enough to try out Launchpad.

Launchpad is available (price £20.00) from either the author:

Dilwyn Jones, 41 Bro Emrys, Tal-y-bont, Bangor, Gwynedd, LL57 3YT, U.K.

email: dilwyn.jones@tesco.net

Payment in Pounds Sterling only

or from:

Q-CELT COMPUTING,

The Falconry, Glenmacnass, Glendalough, Co. Wicklow, Ireland

email: darrenb@esatlink.com (Payment in Pounds Sterling or Euros)

QDT US Edition

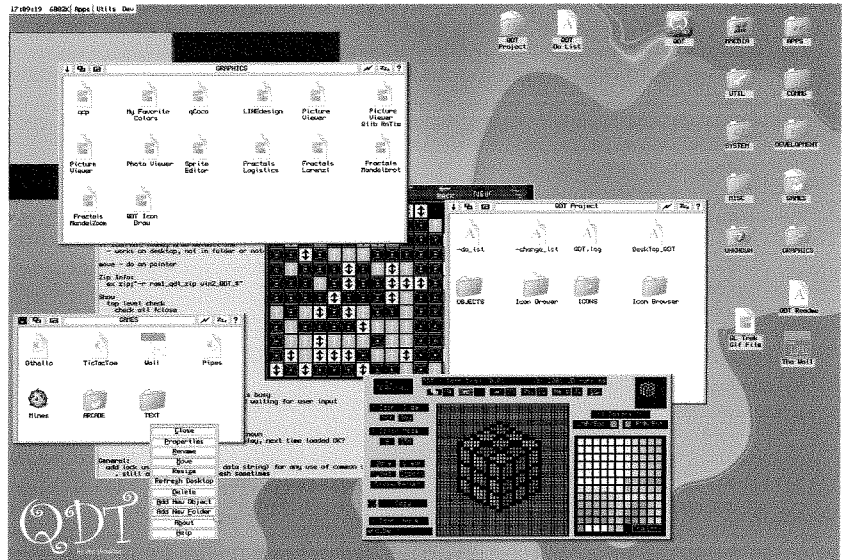
Marcel Kilgus

As Jim Hunkins returned quite ill from the US QL show (an infected spider bite it is rumoured) Jochen asked me to write up a few lines about the latest QDT progress. I am of course in no way qualified to do this, but nonetheless here we go:

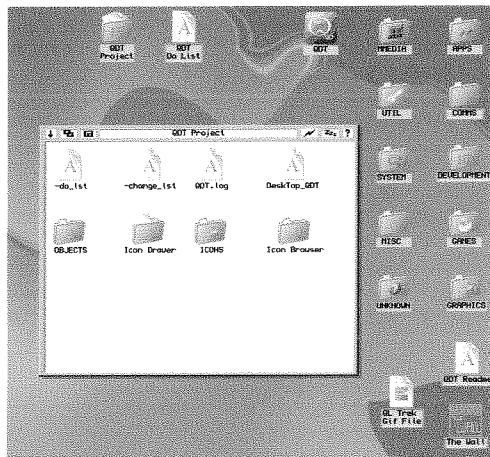
Only one week before the US show I've tested the latest beta version and apart from some minor points and suggestions the biggest request I had for him ran along the lines of "For god's sake, make it more pretty, get rid of the stippled backgrounds and finally use the high colour capabilities I've added to WMAN. I mean, you work for Apple, you of all should know that at this stage of a desktop project 'pretty' is even more important than 'works 100%'".

When we finally met at the show I was seriously baffled by the beautifying QDT had undergone in only one week. Again it is a pity that QL Today can only be printed in gray because the screen shots will certainly not do it justice. So, what was new? First of all I gathered that Jim got contacted by a graphics designer, Thorsten, who has stumbled over the QDT web page and, currently being in between jobs, has volunteered to do some icons for the project. Big optical improvement. Also Jim's demo included a very nice bluish background image and, last but not least, the windows now had a nice gray 3d look to them. All these things combined really made a big difference.

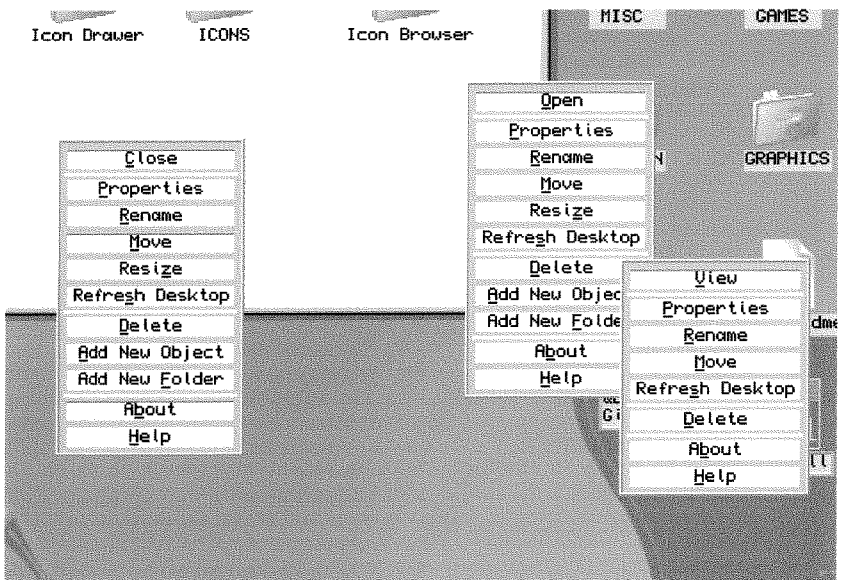
Before all this Jim started the presentation by installing a QDT version from scratch. Let's



Top screenshot:
full screen capture



Middle screenshot:
loser view of new icons
and folder



Bottom screenshot:
sample of three different
context sensitive
menus

just say the installation program alone is now probably much more complex than many actual commercial applications. It offers an expert mode for the more knowledgeable user and also comes with an automatic mode that uses

common defaults for the rest. Apart from copying the program files it also scans the drives for applications to create icons for and even analyses the existing boot file to check whether it includes all needed extensions and

settings. Of course there were also technical improvements which I don't remember in detail right now, so let's just say that there were fewer "not implemented yet" boxes around. Even during Jim's stay in hospital QDT development went ahead (that's what I call dedicated) and I have heard that icons can now be moved with-

in a folder, which is the first step to complete drag and drop support. Looking at the rapid progress one can tell that this is the phase in development which is the most fun for the programmer because real and visual progress can constantly be observed. It reminds me of the excitement I experienced when, for the first time

ever, my 68k emulation core managed to show a flashing SuperBasic cursor. That's when you think "yes, it was worth it". So let's hope that the fun continues and that many future cool features will appear, but also that the project will soon get into a stage that can be released to public so that everybody can enjoy it.

Gee Graphics! (on the QL ?) - part 38

H. L. Schaaf

Gamma

Wondering about Lame curve areas led me to the Gamma function.

An article(1) by J Allard (? any relation to Xbox ?) contained formulae for areas and volumes bounded by Lame curves. Those formulae used a Gamma function, so I began looking for a Gamma algorithm for the QL. That led to an interesting website(2) with a QBASIC program for Gamma, easily converted into QL SuperBasic. What's gamma, and Gamma? Let's stick to the positive integers such as 1, 2, 3, 4, ... and so on. If we multiply them we get $1*2*3*4 = 24$. The exclamation point (bang) is used to indicate this chain multiplication of all the integers from 4 to 1 (or 1 to 4) as $4! = 24$. So 24 contains 1,2,3,4 as factors, and $4!$ is 4 factorial. Five factorial is 120. By definition $0! = 1$. Fancier factorials include the double, triple, hyper and super. You have probably seen algorithms for factorials given as examples to demonstrate recursive programming.

In the year 1768 or so Leonhard Euler took a look at ordinary factorials and asked what values would be appropriate for the factorial of real and ima-

inary numbers such as 1.2, π , $2/3$, etc. What do you think $4.5!$ should be? What about $.5!$, or $1.5!$, or $-2.2!$? Euler developed an integral of the first kind which he referred to as Beta, and an integral of the second kind, which he referred to as Gamma. He also worked out

the slope of the Gamma function at 1 and called it little gamma. He worked out 16 digits and the result, $.57721566490153286\dots$ is now known as the Euler constant, symbolised by the lowercase Greek letter gamma.

Euler is responsible for other mathematical symbols, including lowercase Greek pi, the exponential base e, the imaginary square root of -1 as i, the uppercase Greek sigma for summation, and several others.

Listing Gamma_Allard_bas

```
100 REMark Gamma_Allard_bas
110 REMark H L Schaaf May 3, 2004
120 REMark for GG#38
130 :
140 REMark adapted for QL from:
150 REMark Qbasic Fractorial program
160 REMark 1985-2003 C. D. Wehner
170 :
180 DIM coefft(11)
190 RESTORE 260
200 FOR i = 1 TO 11
210 READ coefft(i)
220 END FOR i
230 REMark enclosing Wehner's 16 digit numbers in quotes
240 REMark gets us an extra digit or so of precision
250 REMark beyond the QL's usual 8 or 9
260 DATA "1", "-.5772102607824947", ".9888923019664764"
270 DATA "-.9054159533130046", ".9671397073277022"
280 DATA "-.9165100019750683", ".7932816063638577"
290 DATA "-.5610449936718201", ".2926044622941572"
300 DATA "-.0966100342863991", ".0148731660765881"
310 :
320 DEFine FuNction Gamma(x)
330 LOCaL ix
340 IF ( (x <= 0) AND (x - INT(x) = 0) ) THEN
350 PRINT #0; ' undefined value ! infinity ? '
360 RETurn 2^2047
370 END IF
380 ix = x - 1 : a = ix - INT(ix) : b = 0 : polynom : updown
390 RETurn b
400 END DEFine
410 :
```


In the year 1814 Adrien-Marie Legendre published more work on the Gamma function, his use of the uppercase Greek letter Gamma set the style. Legendre also worked out extensive tables of the elliptical functions as mentioned in GG 6&7 in 1998. Remember the fun we had with elliptical functions then?

The use of ! (bang) for factorial is credited to Christian Kramp in 1808. Charles Douglas Wehner refers to the Gamma function as 'fractorials', an apt description.

$\Gamma(n) = (n-1)!$ for integers.
 $\Gamma(1) = \Gamma(2) = 1! = 1$.
 $\Gamma(1) = 0! = 1$, $\Gamma(n+1) = n!$, $\Gamma(5) = \text{SQRT}(\text{PI})$.

The Gamma function goes away at zero and negative integer values. The listing will plot the classical Gamma function from -4 to 5.

For simple Lamé curves we can use the listing based on Wehner's program to find the Gamma function needed in Allard's formula for area. The Allard's formula in the listing is based on one quadrant with both major and minor semi-axis = 1. If the axes are not 1, then multiply the Allard value by the product of the semi-axes.

So far, all the references I've found to Lamé curves use the same exponent for both the x and y terms, yet we know we can use mixed exponents to produce ogee curves. Perhaps these compound curves go by another name? By any name they make finding the length and area more challenging, and led us to Gaussian quadrature with help from Legendre.

Still no general solution for the length of Lamé curves, but we do have a few examples with known area and length, namely the astroid, diamond, and circle. The areas and arc lengths given below are for one quadrant, where major and minor semi-axes both have unit dimension.

```

420 DEFine PROCedure polynom
430 LOCAL n
440 FOR n = 11 TO 1 STEP -1
450   b = b * a
460   b = b + coefft(n)
470 NEXT n
480 END DEFine polynom
490 :
500 DEFine PROCedure updown
510 IF a < ix - .5 THEN raise
520 IF a > ix + .5 THEN lower
530 END DEFine updown
540 :
550 DEFine PROCedure raise
560 a = a + 1 : b = b * a : updown
570 END DEFine raise
580 :
590 DEFine PROCedure lower
600 b = b / a : a = a - 1 : updown
610 END DEFine lower
620 :
630 DEFine FuNction Allard(expon)
640 numer = (Gamma((1/expon)+1)^2)
650 denom = Gamma((1/(expon/2))+1)
660 RETurn numer/denom
670 END DEFine
680 :
690 DEFine FuNction Beta(p,q)
700 LOCAL pq, gp, gq, gpq
710 pq = p + q : gp = Gamma(p)
720 gq = Gamma(q) : gpq = Gamma(pq)
730 RETurn (gp*gq)/gpq
740 END DEFine
750 :
760 REMark show the Gamma function plot
770 WTV : PAPER 0 : INK 7 : CSIZE 2,1
780 SCALE 60,-45.5,-30 : CLS
790 AT 0,3: PRINT 'GAMMA Function'
800 CSIZE 0,0
810 FOR i = -4 TO 5
820 LINE i*10, -25 TO i*10, 25
830 CURSOR i*10, -25, -2, 0 :PRINT i
840 END FOR i
850 FOR i = - 20 TO 20 STEP 10
860 LINE -40, i TO 50, i
870 CURSOR -45, i, 0, -4 : PRINT i
880 END FOR i
890 INK 2
900 LINE -40, 0 TO 50, 0 : LINE 0, -20 TO 0, 20
910 INK 4 : POINT -4, Gamma(-4)
930 FOR i = -3.99 TO 5 STEP 1E-2
940 LINE TO i*10, Gamma(i)
950 END FOR i
960 REMark end of listing Gamma_Allard_bas

```

Figure	Lamé exponent	Area of Quadrant	Arc length of Quadrant
astroid	2/3	(3 * PI)/32	1.5
diamond	1	1/2	SQRT(2)
circle	2	PI/4	PI/2

Next time we hope to use Legendre-Gaussian quadrature to find the area bounded by compound Lamé curves.

References:

- (1) "Notes on Squares and Cubes" in *Mathematics Magazine*, September 1964, Vol. 37, No. 4, pp. 210-214 by J. Allard, University of Sherbrooke
- (2) Charles Douglas Wehner's webpage: www.Wehner.org
 Qbasic 'Fractorial' program

The Epson "LQ 300+" Printer

Paul L. Harris

This is a slightly irritating device. It is a 24-pin printer which works, very satisfactorily, from "Quill" and from "Text 87". I have not yet managed to get it to perform from "LineDesign" but that is because I have not used that programme for a long while and have yet to re-learn it.

I have used it successfully off a "black box" with an S.G.C., via Miracle's interface (the parallel lead got left behind) and on which this review is being written, and direct from the parallel port of a Q40. It has a "push-tractor" feed for continuous stationary and will take single sheets of A4, though only when fed with these one at a time. It is also said to print in colour when fitted with the suitable striped ribbon, but I have not tried this, not having the ribbon. The black ribbon is reasonably easy to install.

A great advantage over my "Stylus 740" is that it does not suffer from the cold. If the room where that lives is not kept warm the 740 is very uncooperative. I would be very interested to hear any

suggestions as to the cure of this weakness.

The only irritating aspect of the LQ is the paucity of printed instructions. There is a manual, in eleven languages, but the only one of these I can understand is very brief. A CD comes with the machine, but this, of course, is not readable on a QL. If I can read this without installing the printer on a PC I would like to discover how to use the row of buttons on the front and the significance of the lights on the front panel.

Apart from this irritation, it is a very satisfactory machine.

[As far as we are aware, all EPSON LQ type printers (all dot-matrix) are still compatible with ESC/P2 and therefore can be used with virtually any QL program which supports an EPSON driver. Luckily, all these printers seem to support a parallel port interface.

Would be nice to know if it does colour print with a colour ribbon and SDUMP (which used to be able to print out in colour on older dot matrix printers).

No idea about the "cold" problem - I have not heard this ever before and our EPSON Stylus printers (Colour 740, several Colour 900, Colour 980) work at temperatures between 16 and 36°C - Editor]



JUST WORDS!

The big one...

QL2004

EINDHOVEN

16th October

For more information visit our website:

<http://members.lycos.co.uk/geoffwicks/ql2004.htm>

or use the Just Words! contact information.

Geoff Wicks, 56 Peveril Crescent, West Hallam, Derbyshire DE7 6ND, U.K.

Tel: +44 (0)115 - 930 3713

email: gwicks@beeb.net

Web: <http://members.lycos.co.uk/geoffwicks/justwords.htm>

Bad Boys Rape or We PIN-DOWN?

Al Boehm

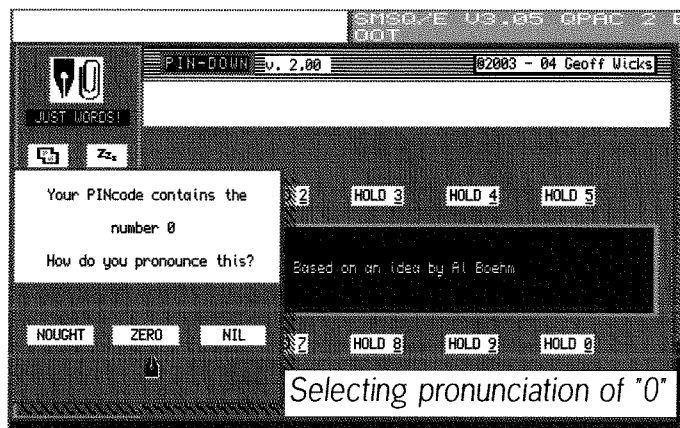
I was at the store ready to buy some groceries with my debit card. All I had to do was punch in my PIN (Personal Identification Number). No go, I had forgotten it and could not find the little piece of paper I had surreptitiously written it on so no one else could figure it out. There has to be a better way. [An aside: Thanks to Spelling Crib or I never would have figured out how to spell surreptitiously!]

How many PINs do you have? For one bank, I have a PIN for the debit card, another for their web site, and a third for the automated telephone. For my other bank, I have two more. Plus one for my insurance web site, another for my mortgage, and on and on. Then there are telephone numbers. And then there is my bicycle lock.

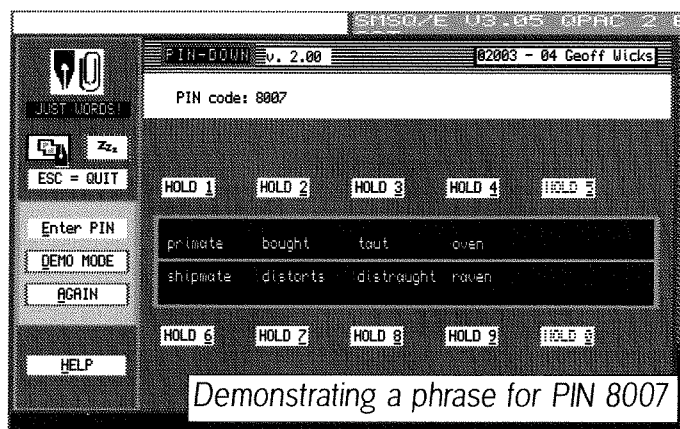
Help! Help!

Can you look at the colors on a resistor and tell how many ohms it has? I can although I have not done it for several decades now. The reason is I was taught a mnemonic some 40 years ago and it has stuck in my head: "Bad Boys Rape Our Young Girls But Violet Gives Willingly" which stands for

Black=0,
Brown=1,
Red=2,
Orange=3,
Yellow=4,
Green=5,
Blue=6,
Violet=7,
Gray=8,
White=9.



Well, if I could remember the color code for resistors for 40 years, there had to be a similar way to remember PINs. My first guess was to try to make a



word using the letter to number relation on the phone dial. There is no letter for 0 or 1 but that could be worked around somehow. However, you would need a dictionary with potential words. It is at that point I thought of Geoff Wicks.

The result is PIN-down, a QL program to rhyme phrases to numbers up to 10 digits long. Right at the start, I must tell you

that Geoff did all the programming. In addition he came across the idea of using rhyming words for numbers. Still I claim I did the hard part. No, not the original idea – that came easily. The hard part was convincing Geoff to do the project. Over many months, I wheedled, I begged, I even threatened, "If

you don't write the program, than I will, and the QL community will be stuck with an inferior program. So there!"

In any case, whether through my nagging or not, Geoff went ahead and wrote PIN-DOWN and the QL community is not stuck with an inferior program.

So what does it do?

Basically you give it a number and it prints a selection of random rhyming phrases. When it starts up, you hit 'Enter PIN' either by pressing E or by moving the mouse over the Enter PIN button. Then enter your PIN. It then prints a random phrase in which each word rhymes with the digits you entered. For example, if you entered 5678 it might print: "Midwife evicts unshaven curate" where Midwife rhymes with 5, evicts with 6, unshaven with 7, and curate with 8.

If you don't think that phrase is memorable, you hit 'Again' either by pressing A or clicking with the mouse, and another random phrase is printed.

Actually when the PIN is five digits or less, two phrases are printed at the same time. Further if you like some words but not the whole phrase, you can hold a given word by hitting 1 for the first word, 2 for the

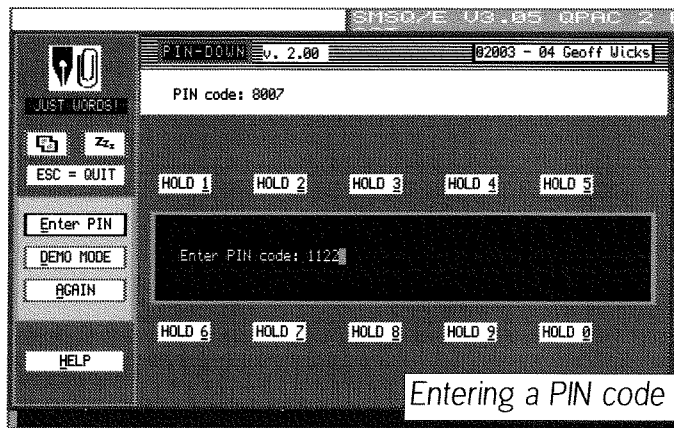
second, and so on, or you can click Hold with the mouse. Those words will not change.

If your PIN contains 0, PIN-DOWN will ask you do you want it to rhyme with naught, zero, or nil.

Once you find a memorable phrase, you can quit or do another PIN. PIN-DOWN is very easy to use. It has a built

in Help file which is hardly needed. Also it comes with a separate text file with instructions which are hardly needed. There is a demo mode which is fun to watch as the random phrases are generated.

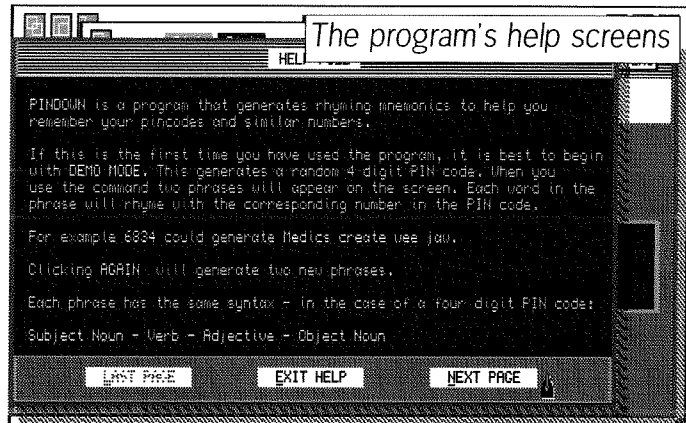
PIN-DOWN requires Toolkit 2 and the PE. It takes up about 53K of memory. It can be moved around on the screen. There is nothing special to set up; just ex flp1_pindown_obj and away it goes. Of course,



you can put it on your harddisk instead of flp1_. You can purchase PIN-DOWN from Just Words for 5 pounds.

People talk about things a PC can do that

the QL can not. Well here is one thing the QL can do that has no counterpart in the PC community. Oh, one last thing, if you really see a need for a new program, nag your favorite QL developer. When all's done, they will thank you.



QL show - Orlando, Florida, USA. May 04

Tony Firshman

The USA is Big Country

I flew to New York a week before the show with bicycle to stay with friends in Larchmont, NY. This is a very posh commuter suburb north of Manhattan. Their second car is a brand new 2-seater 4L Nissan softtop. The runabout is a brand new vast 5.9L Nissan Armada 4x4 - do the Japanese not study European history? Both have very accurate 3D satellite navigation that tells you where you are - saves looking out of the window.

I set up my own wifi access points at their house, and had full access to my system in Tring - pity my email died the

day I left! Had skype conversations with Vic Gerhardi (Rake-well - Z88 man) who lives a mile away from me in Tring. Sound quality was better than from home!

Bought a 17" 2.8GHz Toshiba for worldnews.com - so that was to be a DVD player for the journey. I bought a 140W mains inverter that plugs into a cigarette lighter. (see later) I repaired their bath drainage system, kitchen lamp, 2 doors, and a neighbour's window, broken by their son's baseball attempts. We European's are useful.

Phoebus said that the 'free' AOL for internet at the show was not free. He had a GSM

sim card with free access at weekends, but no suitable phone. I configured my Toshiba Libretto P266 and found that it made IR contact with my phone which dialled - we were in business (in theory).

Bill Cable arranged to pick me up at 2am Thursday morning, and we drove non-stop to Orlando, with a diversion to pick up Phoebus Dokos in Indiana PA. I don't think either of us had realised it would be a 6 hour diversion!

The USA is a big Country.

It is ironic that the road we started from in NY (I 54) goes within a few miles of Orlando!

Bill arrived nice and early in his Saab - I had a passing thought about borrowing the Armada. 11 miles to the gallon though is not good, even at \$2 to the gallon or less.

Phoebus had to get his daughter Mikaela to school

and pack, so we had a 2.5 hour break in Indiana. That is the town near Pittsburgh, not the county! I was getting nervous (sorry Phoebus) about missing my day jaunt with the lads in Orlando - I knew Jochen wanted to start early to get on the rides. I had a shower and managed to get some business done on the internet - a US site was complaining about **worldnews.com**, and I managed to calm them down on the phone (thanks Phoebus).

I plugged in the Toshiba laptop and power supply - the 140w inverter flashed overload and Bill's cigarette socket splitter melted! I then watched Lord of the Rings part 2 (It is fabulous with wide 17" LCD and headphones) on battery power. 2.5 hours later (before the end of the film) the battery died. No problem - I will charge the battery. Nope - still overload! Do Toshiba not realise that one does not always want fast charge? Journey's cinema is over. Still the inverter was useful to charge my mobile phone - which had decided that US roaming would eat its battery in no time at all.

Bill's car (and us) behaved impeccably, despite its 200,000 miles and New Hampshire forest rocks/dust in the engine compartment. Tarmac is a piece of cake for it. The only repair was to patch a cruise control hose. We hit a rock in Road Work which scraped the rim and took a gouge out of a tyre - it survived.

We arrived in Orlando at about 11am Friday - a mere 34 hour journey. One item on Mapquest

was like driving from Brighton to the north of Scotland.

The USA is a Big Country.

We then tried the phone/Toshiba Libretto setup - it connected. We were in business for the show internet - but at 9600 bps!



We then had a Guinness at the local 'Irish pub', lunch, showered and fell asleep at 4pm, having texted Jochen to say we had arrived. The Chilis' sandwiches were giant, and Phoebus had two! I woke up at 9pm to find that no-one had been able to raise us at 7pm.

I joined the rest at Chilis during happy hour (2 drinks for one) to find Roy almost asleep! Sunbathing in Miami is hard work (8-)#

We then went to 'our' Irish pub ('Is there pool' asked Marcel) only to find it was much more expensive than 'their' Irish pub.

Bed again much later, and up early at 8am for the show.

I set up the internet connection (see <http://www.firshman.co.uk/showphotos/USA-2004/index.htm>) and it worked perfectly - all day. Most of the photos were uploaded to my web site on the day.

However being 9600 bps, every time it failed to respond, the cry went out 'Alf - shut down your email'. He was using a web based service that consumed vast bandwidth! The show had non-trader representation from both USA and Europe -

one coming from California which is almost as far as Europe). There were though only about 15 at the show, which was disappointing.

Al Boehm gave a presentation of Just Word's new PIN acronym program PIN-DOWN (described in another article this issue).

Jim Hunkins showed us how near QDT (the new QL front end running under

SMSQ/E) is

to release. This is going to be a terrific product.

I did my usual host of hardware repairs.



We forgot the forum. We adjourned for supper - changed by public demand from Pizza Hut (sorry Al - we are too familiar with that from the UK) and thence to the other Irish pub - which again was serving 2 drinks for the price of one! More Guinness - this is beginning to sound like the last Irish show, Darren. Bill, Phoebus and I set off just after midnight Saturday for the long haul home. We arrived in Indiana at 9pm Sunday.

The USA is a Big Country.

After a quick shower Bill and I set off for Albany, NY, in the wrong direction. We turned round and set off again in the right direction but the wrong road - but it was going to where we wanted. We were both hungry, so we looked for food. After many false sitings (24 hour Dennys where the kitched closed at 10pm, 24 hour 'open' restaurants with lit signs, that were closed etc). We eventually found somewhere at 05:30! Very greasy food, and I joked with the waitress that I wouldn't think it

was such nice food in 30 minutes. Both Bill and I WERE suffering on the rest of the journey!

We arrived in Albany where I



was to catch a Greyhound to Manhattan. Off at exit 4 said Mapquest - but a car would not allow us to change lanes - so we had to go 5 miles further, with a traffic queue on the other carriageway. Fortunately we found the bus terminal with no problem - at 09:30 Monday.

The USA is a Big Country.

I caught a bus to Manhattan, got given an MTA ticket by a passing tourist, asked directions from a passing retired PanAm pilot (Did you realise

that they not only lost their free flights, but their pensions as well? Libya did not pay that) and arrived in Larchmont at 14:30 after 38 hours.

The USA is a Big Country.

I set off for home the following day, with rucksack, 10lb laptop and bicycle. I negotiated MTA to Howard Beach station to find that the free bus direct to the terminals had been replaced by an expensive 'Sky Train' that went very slowly through all the car parks.

Booked in my 48 kg of luggage (still way below the allowed 64kg - how can anyone carry all that!), and went to security. It was only then I realised that I still had my beloved Opinel penknife in my pocket. I hid it under the laptop power supply and it sailed through unnoticed. However the scissors (inside my bandaid strip) were confiscated!

I arrived home without further incident.

Never underestimate the size of the USA.

What Now?

David Denham

Recently, my QL developed a fault. It would run for a while and either freeze or reset itself. I panicked, thinking that either the QL or the Gold Card had come to the end of their respective lives. Fortunately, it was all down to a faulty power supply in the end, and as I had a spare power supply the problem was quickly fixed. But it did make me think about where I go from here if my beloved QL and Gold Card do go to that great computer graveyard in the sky. I realised I would have to start thinking what my next computer system would be. At my time in

life I do not really like major upheavals, but I have to face the facts - my QL is not getting any younger and there is not much by way of new QL hardware being produced.

QL Today is full of mentions of Q60s and emulators and everything else. I had a look at the pros and cons of all these systems and decided to set my thoughts down on paper so that it can also invite thoughts on the subject from fellow readers. I suppose my way forward could be split into three possible groups.

1. Traditional QL hardware - QL with Gold Card, Super Gold Card and so on.
2. Newer QL systems like Aurora and Q60.
3. Emulators - systems like QXL, QPC, QLay and so on.

1. QL-based Systems

The original QL was released back in early 1984 and many of us now use computer systems which are 20 years old. In some ways, this is a recognition that despite the criticisms sometimes aimed at Sir Clive and his computer hardware, the QL has on the whole managed quite well in the long term reliability stakes. Yes, keyboard membranes and microdrives have often been a pain as the inevitable wear and tear sets in, but there are still plenty of people happily managing to use QLs made in the 1980s. Expansion of these systems can be a bit of a problem in that few expansion options are now produced, but it is fairly easy to obtain second hand disk interfaces, hard disk interfaces and serial-parallel conversion printer interfaces. Mouse interfaces (QIMI and Superhermes) and flash memory devices (romdisq) are still being produced and there is still hope that a new Qubide, Ethernet system and Goldfire expansion unit may be manufactured.

2. New Computers

The Aurora replacement motherboard has been available for some time and the Q40 and Q60 systems from Peter Graf and the D&D Systems gave new QL hardware a much needed boost more recently. The Q60 is generally viewed as an excellent and highly desirable super-QL, offering much higher processing speed than you'd get from a traditional QL or Aurora based system, and the possibilities of adding PC-style expansion cards to the Q60 motherboard with suitable software drivers. The Q60 is available as a ready made system complete with everything from hard disks to speakers and high colour display. The downside of the Q60 is summarised in one word - cost. It is an expensive computer to buy! I have also heard rumours that Peter Graf is not enthusiastic about the SMSQ/E operating system, preferring to go the QDOS Classic route. As QDOS Classic is essentially QDOS, this would seem to be a backward step in some ways.

3. Emulators

QXL

If you wish to combine your QL activities with other computing activities, the emulator path might be the way to go. When I suffered QL problems recently, I remembered being offered a very cheap old PC by a friend who was buying a newer more modern PC. I toyed briefly with the idea of getting this PC and trying to buy a second hand QXL card to put into it. The QXL

would run quite fast, pretty independent of the speed of the slow PC and I could perhaps add an external modem to the PC which QL internet, fax and terminal software (soql, qfax and qtpi) could drive, and even on an old, slow PC, Windows should still be able to drive the modem for internet browsing, do basic word processing and so on. The downside of course is that the main computer is something I have little knowledge of. Something goes wrong with the PC, my QL goes down the tube with it and I probably wouldn't have the knowledge to be able to fix any problem greater than a blown fuse in the plug. I feel reasonably confident taking my QL apart, I also feel reasonably confident tackling QL software issues. My experience of PCs is limited to the most common tasks like internet, email and word processing. On the positive side, putting a QXL into an older PC forces me to think and learn more about the PC and Windows and I suppose it could be argued that by adding to my knowledge of the host computer I'd be in effect enhancing the QL experience - learning to use the PC properly helps me get the best out of the QXL and my QLing. If the PC is too old and too basic to run Windows very well, you can just install a copy of MSDOS and in effect use the computer as a QL and ignore the DOS/Windows side of things if you are not interested in the PC side of things. QXL is quite happy on a DOS only system.

QPC2

This is a software based emulator for a PC running Windows, but a very fast emulator. SBASIC (the version of BASIC supplied in SMSQ/E) is a very fast implementation of QL BASIC. QPC2 is pretty well 100% compatible with a QL equipped with SMSQ/E and offers one or two extra luxuries like easy access to PC floppy and hard disks as well as the colour drivers. Downsides of QPC2 are of course cost (apart from QXL it's the most expensive emulator) and the fact that (like all emulators) you are at the mercy of the host computer and operating system. If your PC crashes a lot, there's not a lot QPC2 can do probably. And QPC2 makes certain demands of a computer, it will need Windows 95 or later, a certain amount of memory, Direct-X software on the PC, the right kind of graphics card (no problem in a modern system). If you have an old PC with small memory and primitive displays, you may be better off opting for a QXL card.

Q-EMULATOR

This is a shareware emulator from Daniele Terdina. It runs on either Windows or Apple Mac sys-

tems (two separate versions of Q-Emulator of course). It runs on a wide variety of hardware and is QDOS based, giving good compatibility with traditional QL systems. The speed varies depending on what it's running on and whether or not it's registered. The copies you get before registering the emulator with the author run at a slower speed and are really only good for trying it out, not really fast enough for use as a serious QL emulator. You need a QL ROM image to run this (and most emulators apart from QPC2 have this need) but there is a copy of the Minerva ROM available if you do not have access to a copy of a Sinclair QL ROM. The emulator is not too demanding of the host computer, but obviously emulation speed may depend on the speed of the host computer.

QLAY

This is a freeware emulator for DOS, Windows and Linux platforms. It's free and runs on pretty well anything running DOS, Windows or Linux. Its disk handling is a bit quirky, requiring a disk tools program to copy disks and move QL software into the QLAY environment. It's pretty easy to get going and free, so you may not be too worried about this. There is also a more recent version called QLAY2 from Jimmy Montesinos as well as something called QL2K which I don't know very much about.

UQLX

Richard Zidlicky took the Qemulator sources and produced something called UQLX, a QL emulator for Linux type platforms. It is free, and a very well respected system. It even has a TCP/IP system access for systems where the host operating system has such facilities, meaning that suitable QL software such as the programs from Jonathan Hudson can access email and internet now without having to wait for SOQL to come along to your own particular system. There were rumours of a Windows version of UQLX from America, but I don't know what the situation is with that.

AMIGA

For those QLers who have an interest in Amiga computers, there are two emulators to choose from. The first is the original QDOS4Amiga, and more recently QDOS Classic by Mark Swift. I don't know very much about these emulators to discuss their relative merits.

ATARI

The ST-QL boards are to Atari STs what the QXL is to the PC. The QL board for the Atari has been

around for some time and well supported by Jochen Merz. Again, this is a well respected emulator system, though I don't know if it is still produced and just how many Atari computer systems there are out there these days. Although I've never used it, there's a software emulator called QeM, a freeware QL emulator for Atari systems.

My own preferred route is a traditional QL system or possibly an Aurora system ready built for me. If my budget would run to it, I'd probably consider a Q60.

Is there a future in traditional QL hardware systems? The original Sinclair QL is twenty years old now and after that time I'm sure many people would say it's time it quietly retired from the scene. It may yet be given a new lease of life by new hardware like Goldfire when (if?) that is released. After all, as long as a QL system is reliable and has the facilities you need of it (plenty of memory, enough processing speed, hard disk, printer and so on), many people will be quite happy with such a system if they have no need for a Windows, Mac or Linux system, which is just added complication to some of us.

Software emulator systems are cheap and easy solutions to the need for more than one computer platform. Some people like to have a PC or Mac or Linux system for work-life applications, and a QL for home computing and hobby purposes and using an emulator provides a neat and simple solution to the problem of desk space - there's no need for two physically separate computers on your desk! Emulators are subject to the vagaries of the host computer system though.

I would be very interested to hear how people feel about this - why not write into QL Today and get a debate going!

Good to see that there is still demand for QL Hardware - Nasta should be pleased. It would be good if users who own various systems compared them. We don't need the "ours is faster" or so, I believe - a rather more useful look into what is going to happen in a months, a years or whatever time related to the current situation would probably be helpful to users faced with the problem that their QL hardware died. It is probably as hard getting an ATARI repaired as it is to get a QL repaired, for example. And how about if a Q40/Q60 hardware problem appears? Please send us your views!
-Editor

Programming in Assembler: Part 10 – Linked Lists

Norman Dunbar

It's been a while since we last took a look at something other than QLTDIs, so I decided to have a look at linked lists this time around. The masses of typing we did in the last QLTDIs article has broken the assembler. When it assembles, pass one works ok, but pass two just dies. I've informed George of this slight problem and knowing him, he'll have it fixed by the time you read this. Unfortunately, I don't have a working version yet, so I'll have to do something else.

On the subject of my most recent typings, I had a couple of emails from Hugh Rooms on the subject of my inability to figure out how to decode a register list in code, rather than by using a large table as I ended up doing – and breaking the assembler to boot.

Hugh reminded me of an algorithm known as a Finite State Machine (or Finite State Automaton as I was taught) and he gave me a possible working FSM on decoding the register list. Hugh's method has one small problem which I'm not too happy with, so I have not reproduced it here – I'm working on it though, and on some example code he also supplied, to make the decoding work 'correctly'.

The problem is simple, Hugh's code produces this

```
MOVEM.L D2/D3–D5,–(A7)
```

instead of my preferred decoding of:

```
MOVEM.L D2–D5,–(A7)
```

However, Hugh's algorithm is so simple, it beggars belief and I'm very grateful to him for sharing it with me. I shall present a slightly amended version of Hugh's method in a future article. In this article, I'm going to ramble on and on and on about the subject of linked lists.

Linked lists are used within QDOSMSQ to hold details of the directory devices installed on the system, interrupt routines and so on, but what are they exactly?

Imagine that you are writing a program, and you decide that you need some storage for some data, let's say a list of people's names and addresses. So, how about an array?

Well, the problem with that is how many entries are you going to allow? If you don't allow enough entries, you won't have much of an address book. If you have too many entries then you are wasting space. If you sell the program, or give it away, then you need to consider the needs of people other than yourself – some will need a few entries and others, much more. How do you cope?

Well, a linked list could be the answer. You start off with no storage defined at all, except for a single, maybe two, variables which hold the address in memory of the beginning (and maybe the end) of your list of addresses. As you add new contacts to your address book, each one is created at some 'random' location in memory and linked into your existing list of contacts. Hence, you have a linked list.

In a linked list, each entry is called a node, and the pointer to the very first entry in the list is known as the root node.

In memory an array, of 10 entries of 100 byte long strings, is consecutive. Don't forget the strings have a word at the start defining their length, so each entry is actually 102 bytes long. If the first entry is located at address 1000 then the next entry is at address 1102, the next at 1204 and so on. There are no gaps between entries and you can quickly calculate the start address of any particular entry as $1000 + (\text{INDEX} * 104)$ where INDEX is the entry you are looking for, starting at zero.

In a linked list, the nodes are all over the place, the first might be at address 1000, the second at 2000, the third at 1200 and so on. There is no logical order to the locations and you cannot calculate the address of a particular node using any formula as you can with arrays.

What you can do, however, is store away the address of the first node in a special node known as the root node, and from that, you can navigate along the list from start to finish by finding the address of the next node from the data stored in the individual nodes. Our 100 byte long strings would be 106 bytes allowing 4 bytes to store the memory address of the 'next' entry in the list and the obligatory 2 byte length word. However, think about that 102 bytes in each entry of the array – you might not need all 102 bytes. In our linked list, each node will have 4 bytes for the pointer and only as much space as is required by the data, so each node need not be 102 + 4 bytes long. Another saving over the array.

A linked list can be thought of like an old program on UK TV, Treasure Hunt, where Aneka Rice used to zoom around the country in a helicopter picking up clues in one location which told her where to go for the next clue and so on, until she found the 'treasure' at the end of the list of clues. This is exactly what a linked list is.

If we have a node in our list defined as follows, then we can see how it looks in memory below. Each node in the list will look like this:

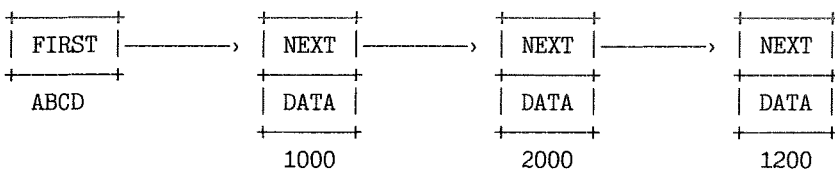
```

+-----+ 4 bytes at the start of each node hold the address of the NEXT node in the
| NEXT | list.
+-----+
| DATA | The data bytes can be any size and hold any data we like. It can even hold a
+-----+ linked list itself. The possibilities are endless.

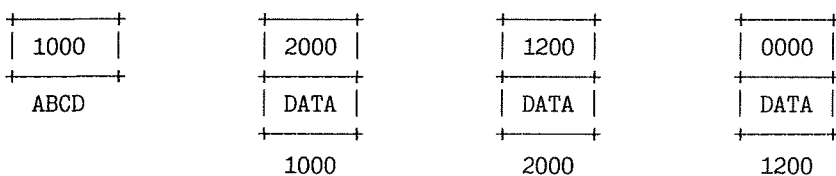
```

The root node, as mentioned above, is special. It has no data part, only the pointer part.

The conceptual layout in memory is a bit like this (using the addresses mentioned above and assuming the root node lives at address \$ABCD):



In physical terms, there are no handy arrows. Using real values as described above in the pointer locations, it would look like this:



You can see the address of the following node in each node's first 4 bytes above. The address of FIRST is actually somewhere in your program and your program only needs to allocate storage for the 4 bytes it takes to hold the address of the initial node in the list. FIRST is, of course, the root node of the list.

You must store the value zero in there before you go off adding nodes, you'll see this reason why below in the code to add a node.

Adding Nodes

Adding a new node is simple, you allocate it on the heap, fill in the data part and add it to the front of the list. It is far easier to add a node at the start – address 1000 in the above example – than to have to work through the entire list to find the current end, and then add it there. This method takes longer and longer to carry out as you add extra entries to the list. Adding at the start of the list takes the same time regardless of how many entries are in the list.

As you add each node to the list, you copy the value in FIRST into the new node's NEXT pointer and put the address of the new node into FIRST. Sounds complicated, but here it is in code. If we assume that A0.L has the address of FIRST and that A1.L has the address of the node to be inserted into the list, as contrivingly demonstrated by these two lines:

```
Prelude    lea FIRST,a0          ; Pointer to storage of first node address
           lea NewEntry,a1      ; Address of new node
```

Then adding a new node to a list is as simple as this :

```
AddNode   move.l (a0),(a1)    ; Save current first node in new node's NEXT area
           move.l a1,(a0)      ; Store address of new node in FIRST storage area
           rts
```

Nothing to it. The new node is always added at the start of the list, so the value in FIRST always points to the most recently added node. As you need to have zero in the NEXT pointer of the final node in the list, you can see why it was important to initialise the value in your programs FIRST variable to zero before adding any nodes. If you didn't have zero, you'd never know when the list was finished.

One thing, you don't want to allow the user to add the root node to its own list at any time, so best change the above code to prevent this from happening.

```
AddNode   cmpa.l a0,a1        ; Don't allow the root node to be added again
           beq.s  AddExit      ; Bale out quietly if attempted
           move.l (a0),(a1)    ; Save current first node in new node's NEXT area
           move.l a1,(a0)      ; Store address of new node in FIRST storage area
AddExit    rts
```

Another problem is when you try to add a node that is already there. So to be really careful, you could call the FindNode routine (coming soon – have patience!) prior to adding it in. However, as this scans the entire list until it finds or doesn't find the new node, it could add quite a lot of time to the simple exercise of adding a new node.

If you wrote the program, and you are allocating nodes on the heap each time, then don't bother attempting to find the node in the list before you add it.

Deleting Nodes

Deleting a node is slightly more difficult. The node to be deleted could be anywhere in the list, or not even in the list. How to find the correct node is the main problem. However, for the same of argument, assume that we have the node address to be removed in A1.L and the address of FIRST in A0.L after a successful 'find' operation, then removing the node at A1.L requires that we must navigate the list, as in the following explanation.

We must navigate the list because we don't know where in memory the node prior to the one we wish to delete is. We need to find it, because it has a NEXT pointer holding the address of the 'deleted' node and this has to be changed or we lose everything in the list after the deleted node.

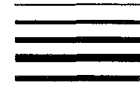
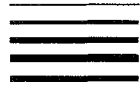
As ever, the value in the NEXT area of the very last node in the list is always zero. That way, we know when we have hit the end of the list. Here's the pseudo code to delete the node at A1.L from the list beginning at (A0.L)

If the node to be deleted is the root node (the list pointer in A0) then don't allow it to be deleted.

Start of the main loop.

If the value stored in the address that A0.L points to is equal to zero, we have been passed an incorrect node address to delete. Exit from the loop with an error.

Q U A N T A



Independent QL Users Group

World-wide Membership is by subscription only,
offering the following benefits:

Bimonthly Newsletter – up to 40 pages

Massive Software Library – All Free !

Free Helpline and Workshops

Regional Sub-Groups. One near you?

Advice on Software and Hardware problems

Subscription just £14 for UK members

Overseas subscription £17

Barclaycard: Visa: Access: MasterCard

Now in our Twenty First Year

Further details from the Membership Secretary

**John Gilpin, 181, Urmston Lane
Stretford, Manchester, M32 9EH**

Tel. +44 (0) 161 865 2872

or

Visit the Quanta Web Site

<http://www.quanta.uni.cc>

E-mail: quanta_membership@uk2.net

If the value stored in the address that A0.L points to is not the same as the value in A1.L then copy the value in the address that A1.L points to into A0 and restart the main loop. Basically we have replaced the address in A0 with the NEXT address from the node we were just looking at.

If the value stored in the address that A0.L points to is equal to the value in A1.L then we have found the node PRIOR to the node we wish to delete and so the node we are looking at has to have the NEXT address updated to bypass the node we wish to delete so that it now points to the NEXT address which is currently stored in the node we are deleting. Exit from the loop with no errors.

End of main loop.

That's the pseudo code, here's the actual code. Using the same preliminary stuff as above to sort out initial values of A0.L and A1.L and a little bit extra to show whether errors have been detected or not, we begin with this:

```
Prelude    lea FIRST,a0          ; Pointer to storage of first node address
           lea OldNode,a1      ; Address of node to delete
           moveq #ERR_EF,d0    ; End of file = node not found = error
```

Now, here's the actual code to find and remove the requested node.

```
DelNode    cmpa.l a0,a1        ; Don't allow the root node to be deleted
           beq.s DelExit      ; Bale out with error if attempted

DelLoop    cmp.l #0,(a0)       ; Reached the end yet ?
           beq.s DelExit      ; Yes, node not found, exit with error

           cmp.l (a0),a1       ; Found the PRIOR node yet ?
           bne.s DelNext      ; No, skip over the deletion code & try again

DelFound   move.l (a1),(a0)    ; PRIOR node's NEXT = the deleted node's NEXT address
           moveq #0,d0         ; Node found and deleted ok
           bra.s DelExit      ; Bale out with no errors

DelNext    move.l (a0),a0      ; A0 now holds the NEXT node in the list
           bra.s DelLoop      ; Go around again

DelExit    tst.l d0           ; Set zero flag for success, unset for error
           rts
```

The above code returns with the Z flag set if the node was deleted from the list, and unset if the node was not in the list. This allows the calling code to handle the errors correctly.

Finding Nodes

The first thing you must do when deleting a node is to actually find it. The code above assumes that A1 holds a valid node address in the list defined by A0. Having said that, the code is robust enough to know that programmers make errors and it can handle the problem of a node address being passed which is not in the list by virtue of the fact that it scans the list until it finds the node prior to the one we wish to delete. It has to work that way because we need to adjust the NEXT pointer in the prior node to point past the deleted node to its NEXT node – if you catch my drift?

The code to find a node in a list is dependant on the sort of data stored in each node. If you store strings, the some form of string comparison routine needs to be built in – does it compare on an equality basis ('AAA' = 'AAA') or nearly equal basis ('AAA' == 'aaa') and so on. You can use the built in QDOSMSQ routines to do the comparisons.

If the data in the nodes are numbers (integers of word or long length) then you can compare them directly. If they are QDOSMSQ floating point format numbers, you can use the built in arithmetic routines

to compare them. Regardless of which method is used, you need to write your own code to compare two nodes, or a node and a value so that the find routine knows when it has found the correct entry.

Of course, it is quite simple to build a FindNode routine which doesn't know or care what sort of data the individual nodes contain, provided it is passed the address of a routine which does know and care. If the specification for said routine requires the Z flag to be set for found and unset for not found, it could look something like the following pseudo code.

Assume that A0.L holds the address of FIRST, A1.L holds a pointer to a routine which compares the node with a given value and A2.L holds a pointer to that value. The data that A2.L points to can be anything, the routine at (A1.L) does the working out, our FindNode simply calls the routine once for each node in the list until such time as it gets a set Z flag on return. The comparison routine gets passed a node address in A3.L.

Start of the main loop.

If the value stored in the address that A0.L points to is equal to zero, we have not found a node with the desired value. Exit the main loop with a NOT FOUND error.

Copy the address at (A0.L) into A3 and call the routine to compare data. If it returns with the Z flag set, the address in (A0.L) is the address of the node prior to the node we were looking for, however, the address in A3.L is the address of our required node as it is taken from the NEXT pointer. Remember, we passed the NEXT address (A0.L) over to the routine, not the address of THIS node - A0.L. Exit from the loop with the Z flag set to indicate a found node.

Copy the NEXT address from the node we are looking at into A0.L and go back to the start of the loop.

End of main loop.

And here's the real code to do the finding for use. As ever, we start off with some contrived values.

```
Prelude    lea FIRST,a0          ; Pointer to storage of first node address
           lea Compare,a1      ; Address of node comparison routine
           lea Required,a2     ; Address of the data we are looking for
           moveq #ERR_NF,d0    ; Node not found = error
```

Now, here's the actual code to find a node in the list which holds the required value.

```
FindNode   cmp.l #0,(a0)       ; Reached the end yet ?
           beq.s DelExit      ; Yes, node not found, exit with error

           move.l (a0),a3      ; Fetch the NEXT node address into A3.L
           jsr (a1)           ; And jump into the comparison routine
           beq.s FindExit     ; Looks like we found our node

FindNext   move.l (a0),a0      ; A0 now holds the NEXT node in the list
           bra.s FindNode     ; Go around again

FindExit   tst.l d0           ; Set zero flag for success, unset for error
           rts
```

The following is an example of a compare routine to look at a long word of data in the node at (A3.L) and see if it is equal to the long word of data stored at (A2.L). Don't forget, the comparison routine must preserve A0, A1, A2 and D0 or it will all go horribly wrong. The following routine does exactly that, by the simple method of not actually using those registers at all!

```
NData     equ 4                ; Offset form start of node to the data part

Compare   cmp.l NData(A3),(A2) ; Is the data in the node = the value we want?
           rts                 ; Exit with Z set if so, unset otherwise.
```

If an attempt is made to 'find' the root node, then it will fail.

So there you have three short but extremely powerful routines which make linked lists possible. At this point I have to mention that there are actually routines built into QDOSMSQ to do exactly the same work as the AddNode and DelNode routines above, but there is nothing like FindNode – which is a shame. However, you now know how to build linked lists and add and delete nodes. You also know how to find an entry in a linked list so that you can process it in some way.

The Code Wrapper

Putting all of the above together and tying in some extras to allocate nodes etc, here is a small, but perfectly formed program to create a linked list. The following is a wrapper that we shall use to demonstrate first the single linked lists as explained above. Later on, when other types of linked list are explained, we shall drop in only the code we need for the demo

```
* =====
* A test harness 'job' for our linked lists code. What's the point of all
* the explanations if you can't test the code ?
*
* This code is simply a wrapper to allow different demos to be 'slotted'
* in to demonstrate the real code in the article, as opposed to the job
* code.
*
* The code being demonstrated is located at DEMO below. As new demos are
* required, only that bit should (!) need changing.
* =====

* -----
* These are offsets from the start of the job's dataspace where working
* variable are stored. The dataspace is held at (A4) in the job's code.
* -----
con_id      equ    0          ; Id for title channel
con_id2     equ    4          ; Id for main output

* -----
* These are simply user friendly names instead of numbers for various
* bits and bobs, colours etc.
* -----
black       equ    0          ; Colour code for mode 4 black
red         equ    2          ; Red
green       equ    4          ; Green
white       equ    7          ; White
linefeed    equ    10         ; Linefeed character
oops        equ    -1         ; General error code for sub-routines
err_nc      equ    -1         ; NOT COMPLETE error code

* -----
* Constants for use with job control commands. (It doesn't matter if I
* have two names with the same value ! )
* -----
infinite    equ    -1         ; Infinite timeout
me          equ    -1         ; Id for 'this' job

* -----
* Code starts here.
* -----
start       bra.s   LinkList   ; 2 bytes short jump
           dc.l    0           ; 4 bytes padding
           dc.w    $4afb       ; Job identifier must be at location 6
           dc.w    11          ; Bytes in job's name
           dc.b    'LinkedLists',0 ; Bytes of job's name plus padding

LinkList    adda.l   a6,a4      ; A4.L = start of dataspace
           bsr     Mode4       ; Set the screen mode
           bsr     Title       ; Open the title window
```

```

        bsr      Output          ; Open the output window
        bsr      Headings       ; Display headings
        bsr      Demo           ; Do the demo code
        bsr      Finished       ; Advise user that we are done

* -----
* Code ends here.
* -----
all_done  moveq   #mt_frjob,d0    ; Force Remove a job
          moveq   #me,d1         ; The current job
          move.l  d0,d3          ; Any error code to send to Superbasic
          trap    #1             ; Kill this job, its channels and its memory

          bra.s   all_done       ; Should never get here - sanity check !

* -----
* The DEMO code starts here.
* -----
Demo      rts

* -----
* The DEMO code ends here.
* -----

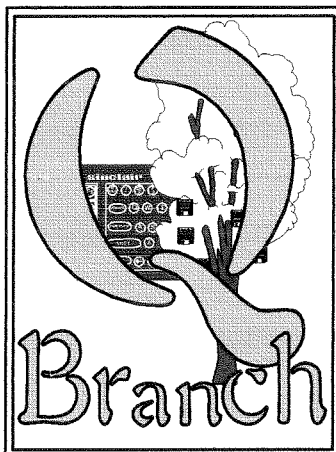
* -----
* Set mode 4 if not already set. Do not change from TV to monitor or
* vice versa. We must preserve the display type if we reset the mode.
* -----
Mode4     moveq   #mt_dmode,d0
          moveq   #-1,d1         ; Read current mode
          moveq   #-1,d2         ; Read current display type
          trap    #1             ; Do it
          tst.l   d0             ; Did it work ?
          bne     all_done       ; No, bale out, cannot continue

          tst.b   d1             ; 0 in D1.B = Mode 4
          beq.s   ModeExit       ; No need to set mode 4
          moveq   #mt_dmode,d0
          clr.l   d1             ; We need mode 4
          trap    #1             ; Set mode 4 (d2 = display type)
          tst.l   d0             ; Check it
          bne     all_done       ; Bale out if errors detected
ModeExit  rts                  ; Done.

* -----
* Mode 4 is current mode. Open the title window at the top of the screen.
* -----
Title     lea     con_def,a1     ; Window definition
          movea.w ut_con,a2      ; Utility to define a window etc
          jsr     (a2)           ; Do it
          tst.l   d0             ; Did it work ok ?
          bne     all_done       ; No, exit program
          move.l  a0,con_id(a4)   ; Store console id for title channel
          rts

* -----
* Definition for title window channel
* -----
con_def   dc.b    red           ; Border colour
          dc.b    1             ; Border width
          dc.b    white        ; Paper/strip colour
          dc.b    black        ; Ink colour
          dc.w    448           ; Width
          dc.w    24            ; Height
          dc.w    32            ; Start position x
          dc.w    16            ; Start position y

```



Text 87

£ 79.00

Typset 94 £ 29.00
 Fountext 94 £ 39.00
 2488 drivers £ 29.00
 Epson ESC/P2 drivers
 £ 26.00

Text 87
 High Colour
 Patch
 Available Now
£9.00

PROGRAMMING

QD 2003	£ 49.00
QD + QBasic	£ 63.00
QD + Qliberator + QBasic	£ 104.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 32.00
Easyprtr pt 1 & 2 (together)	£ 30.00
Easyprtr pt 3 (C library)	£ 14.00
QMake	£ 18.00
QMon / JMon	£ 22.00
Basic Linker	£ 22.00
DISA 3	£ 34.00
QMenu	£ 16.00

Important Notice!

From now on all software supplied by Q Branch will be on HD disks. DD disks are becoming very hard to source and the number I have left here is dwindling fast. If you need the software to be on a DD disk then please state that when ordering. I will save the remaining DD disks for those people who need them.

QDT is definitely coming soon. I am using aa alpha test version of it here at the Q Branch HQ and it works very well indeed. You can see it demonstrated at QL shows this year.

Available Now!

High Colour Drivers for Gold / Super Gold Card SMSQ/E
 (Aurora only - Send old master disk as proof of purchase)

£25.00

QL Keyboard Membranes

Available Now !

£ 17.50 ea + P&P

These programs have been updated to High Colour versions:-

QMake
 WinEd
 FiFi II
 QPAC 1
 QPAC 2
 QSUP

<<<<< SuQcess >>>>>
 Please send old master disk and £ 1.60 to cover postage and copying fee.

UTILITIES

FiFi 2	£ 21.00
QSup	£ 30.00
QSpread 2003	£ 51.00
Cueshell 2	£ 15.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 22.00
QPAC 2	£ 42.00
QTYP 2	£ 31.00
QLQ	£ 30.00

These Programs are paid Upgrades:

QSpread 2001 to 2003	£ 10.50
QD 98 to 2003	£ 10.50
QPC2	-
update from v3.xx	Free
update from v2	£ 13.90
update from v1	£ 34.00

Return Master Disk With Order

Tel +44 (0) 1273-386030 fax +44 (0) 1273-430501

Mobile +44 (0) 7836-745501

email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk

Q Branch

*Feeling out on a limb?
Reach out for Q Branch.
Suppliers of Quality QDOS/SMSQ products
Hardware and Software.*

20 LOCKS HILL, PORTSLADE,
E. SUSSEX. BN41 2LB. UK.

ProWesS

ProWesS (now free !)	£ 1.60
DATAdesign	£ 20.00
Fontutils	£ 28.00
File Search	£ 11.00
PFlist	£ 11.00
Dilwyn's Fontpack	£ Call
LINEdesign v 2.16	£ 22.00
PWfile	£ 17.50

Paragraph

P The ProWesS word processor
 Demo version £ 1.50 + postage
 Full Registered version £ 18.00

Hardware

Second user items are are sometimes available from us.
Call for details.

QXL II	£ 40.00*
Recycled superHermes	£ 65.00 *
R'cycld SuperGold Card	£ 110.00 *
Recycled Gold Card	£ 45.00 *
Recycled Aurora	£ 65.00*
Qubide	£ 45.00*
Qplane	£ 5.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
'Arfa Braquet'	£ 8.00
'Son of Braquet'	£ 18.00
The 'Braquet'	£ 16.00
MC plate	£ 6.50

* when available.

Call for details

Qubide upgrades to version 2.01 £ 8.00

Gold Card / Atari / QXL Version
Version 3.06 out now!

Now Only £ 32.00

SMSQ/E

Aurora High Colour
version available now
upgrade £ 21.00

QPC 2 v3.13 - full colour version!

Special offer!
Get Cueshell for only £15.00 with any copy of SMSQ/E
or QPC 2 upgrade!

Now Only £ 65.00

Q Branch Programs

The Knight Safe 3	£ 35.00	Q - Route v1.08C	£ 25.00
upgrades from previous versions	£ 5.00	Route finding programme	
Q - Count	£ 25.00	The Fractal Collection	£ 35.00
Pointer driven home accounting			

Post and Packing is included with Software Prices
Unless Otherwise Stated.



We can accept payment by Visa, Mastercard and Switch. We also accept cheques drawn on a UK Bank in Sterling.

You can also pay directly into our bank in Euros.

Barclays Bank Acc# 79016888 sort code :20-71-02 Name :QBranch)



Q Branch

```

* -----
* Open the output window underneath the title one.
* -----
Output    lea      con_def2,a1      ; Output window definition
          movea.w ut_con,a2      ; Utility again
          jsr      (a2)           ; Do it
          tst.l   d0             ; Did it work ?
          bne     all_done       ; No, exit routine
          move.l  a0,con_id2(a4) ; Store console id for output channel

          moveq   #0,d0          ; No errors detected
          rts

```

```

* -----
* Definition for output window channel
* -----

```

```

con_def2  dc.b    red            ; Border colour
          dc.b    1             ; Border width
          dc.b    white         ; Paper/strip colour
          dc.b    black        ; Ink colour
          dc.w    448           ; Width
          dc.w    200           ; Height
          dc.w    32            ; X org
          dc.w    40            ; Y org

```

```

* -----
* Print the headings
* -----

```

```

headings  movea.l  con_id(a4),a0  ; Title channel id
          bsr.s   cls             ; Clear screen
          lea     mes_title,a1    ; Title string
          bsr.s   prompt         ; Print title string
          rts

mes_title  dc.w    mes_end-mes_title-2
          dc.b    'Single Linked Lists'
mes_end    equ     *

```

```

* -----
* Sign off message
* -----

```

```

Finished  movea.l  con_id2(a4),a0  ; Title channel id
          bsr.s   cls             ; Clear screen
          lea     end_title,a1     ; Title string
          bsr.s   prompt         ; Print title string
          bsr.s   input          ; Wait for ENTER
          rts

end_title  dc.w    end_end-end_title-2
          dc.b    linefeed,linefeed,'Press ENTER to quit : '
end_end    equ     *

```

```

* =====
* CLS :
* =====
* 1. Clear the (screen) channel whose id is in A0.
* =====
cls        moveq   #sd_clear,d0    ; CLS
          moveq   #infinite,d3    ; Infinite timeout
          trap    #3              ; CLS title window
          rts

```

```

* =====
* Prompt :
* =====
* 1. Print the string at (A1) to the channel in A0.
*
* Z set if all ok, unset if not.
* =====

```



```

prompt    movea.w ut_mtext,a2        ; Print a string utility
          jsr    (a2)                ; Print it
          tst.l  d0                  ; Check for errors
          rts

* =====
* Input :
* =====
* Wait for user input from the channel id in A0.
*
* Returns the input length (not counting the ENTER character) in D1.W
* Returns the address of the first character in the buffer in A1.L
* Preserves the channel id in A0.L
* Z set if all ok, unset if not.
* =====
input     lea    buffer+2,a1         ; Our buffer address plus 2
          move.l a1,-(a7)           ; Save it on the stack
          moveq  #io_fline,d0       ; Input some bytes (inc linefeed)
          moveq  #60,d2              ; Buffer size maximum
          moveq  #infinite,d3       ; Infinite timeout
          trap   #3

          move.l (a7)+,a1           ; Restore buffer pointer
          subq.w #1,d1              ; Subtract the linefeed character
          move.w d1,-2(a1)          ; Store length in buffer
          tst.l  d0                  ; Did it all work ?
          rts

buffer    ds.w    31                ; 60 chars for input plus 1 word for size.

* =====
* hex_l :
* =====
* Convert a 4 byte value in D4.L to Hex in a buffer. Use the input
* buffer for the output and DOES NOT store the length word !
*
* Expects D4.L to hold the value.
* =====
hex_l     swap   d4                  ; $ABCD -> $CDAB in D4
          bsr.s  hex_w              ; Convert the $AB part first
          swap   d4                  ; $CDAB -> $ABCD again
*
          drop  into hex_w to convert the $CD part

* =====
* hex_w :
* =====
* Convert a 2 byte value in D4.W to Hex in a buffer.
*
* Expects D4.W to hold the value.
* Expects A1.L to point at the buffer.
* =====
hex_w     ror.w  #8,d4              ; $DE -> $ED in D4
          bsr.s  hex_b              ; Convert the $D part first
          rol.w  #8,d4              ; $ED -> $DE again
*
          drop  into hex_b to convert the $E part

* =====
* hex_b :
* =====
* Convert a 1 byte value in D4.B to Hex in a buffer.
*
* Expects D4.B to hold the value.
* Expects A1.L to point at the buffer.
* =====
hex_b     ror.b  #4,d4              ; Swap lower and higer nibbles
          bsr.s  hex_nibble         ; Print high nibble first
          rol.b  #4,d4              ; Swap back again
*
          drop  into hex_nibble to print the lower nibble

* =====
* hex_nibble :

```

```

* =====
* Convert a 4 bit value in D4.B to Hex in a buffer.
*
* Expects D4.B to hold the value.
* Expects A1.L to point at the buffer.
* =====
hex_nibble  move.b  d4,-(a7)      ; Save value in both nibbles
            andi.b  #$0f,d4      ; D4.B now = 0 to 15
            addi.b  #'0',d4      ; Now = '0' to '?' (ascii only)
            cmpi.b  #'9',d4      ; Is this a digit ?
            bls.s   nib_digit    ; Yes
            addi.b  #7,d4        ; Add offset to UPPERCASE letters

nib_digit   move.b  d4,(a1)+     ; Store in buffer
            move.b  (a7)+,d4     ; Restore original value
            rts

* =====
* print_hex :
* =====
* Convert D4 into 8 hex characters, then print it to the channel in A0.L
*
* Expects D4.L to hold the value.
* Expects A0.L to hold the channel id.
* =====
print_hex   lea     buffer,a1    ; Output buffer for address
            move.w  #8,(a1)+     ; We know the result is 8 bytes
            bsr    hex_l        ; Convert 4 bytes to text
            lea    buffer,a1    ; Text to print
            bsr    prompt      ; Print it
            rts                ; All done. (Error code in D0)

* =====
* End of test harness
* =====

```

Single Linked List Demo Code

The above code does absolutely nothing, but if you assemble it and exec the resulting file, you should see a pair of windows one with a message 'Single Linked Lists' and a prompt in the other to 'Press ENTER to quit'. Once you press the ENTER key, the job will finish. So far so good.

The reason that it does nothing is shown below :

```

* =====
* The DEMO code starts here.
* =====
Demo      rts

* =====
* The DEMO code ends here.
* =====

```

The code at Demo, does nothing but return to the caller. Our linked list code will be slotted in to replace the single line of code above.

To demonstrate single linked lists, we need only add some code to replace the single line above. Unfortunately, due to the size that this article is growing, you'll have to wait until next time to see the code, however, why not try creating some linked lists of your own in the meantime?

Problem Areas

The above description, and code, is for a Single Linked List, so called because there is a single link in each node which points to the next entry in the list. This is simple to code up – as we have seen – and is fairly simple to understand, at least it is if I've done my job correctly.

The problem with a linked list created in the above fashion is that you always have to scan the list from start to some undetermined entry when you want to delete a node. And this can add serious delays to the processing of your application when a lot of nodes have to be traversed each time you need to delete one.

There is an answer, Doubly Linked Lists. You will find it in the next issue.

Using a CD-ROM on the Q40

Timothy Swenson

A few years back, Thierry Godefroy wrote some drivers that allow a CD-ROM to be used on either a Q40 or systems with QUBIDE. After getting my Q40 back up and running, one of the first things I had to do was to try these drivers out.

By themselves, the drivers are pretty useless to the end user. The drivers provide the underlying mechanism for the Q40 to talk to the CD-ROM. The reason I say drivers, is that Thierry has written the code in two pieces because the drivers handle two different levels of a protocol.

The ATAPI driver handles the ATAPI protocol. This is the protocol that is used to talk to the CD-ROM. This driver comes in two parts, one designed for the Q40 and one for QUBIDE. Because these devices are physically different, a different driver had to be written for each. The ATAPI driver provides a number of Sbasic routines for the ATAPI protocol. The documentation discusses all of the commands in the protocol. All of this is nice, but the general user can ignore it.

The CD-ROM driver takes higher level commands and converts them to ATAPI commands that will be sent to the CD-ROM via the ATAPI driver. This driver also loads a number of Sbasic commands. The two most usefull are CD_EJECT and CD_LOAD. CD_EJECT will cause the CD-ROM to pop out it's tray. CD_LOAD will cause the CD-ROM to retract the tray. It also has the CDR_DRIVE command that I will explain later.

So, we have an ATAPI driver that can send ATAPI commands to the CD-ROM. We have the CD-ROM driver that sends higher level commands to the CD-ROM. The next step is how to actually read the CD-ROM. Most CD-ROM's are written in a format called ISO 9660. This is essentially a file system format like MS-DOS's FAT, or Windows NTFS. There is no current driver that will handle the ISO 9660 format so that we can use SMSQ/E commands, like WDIR or COPY, to read the CD-ROM.

So, Duncan Neithercut has stepped in and written a menu-based program, called QCD-EZE, for browsing the ISO 9660 CD-ROM. The program will allow you to browse the CD-ROM and copy files to RAM or the hard drive. With QCD-EZE I have all I need to use CD-ROM's on my Q40.

Before I can go to far, I need to get the software. The CDRM drivers from Thierry are available from his web page in the file ATAPICD.ZIP From www.Q40.de web site, I found QCD-EZE (QCDEZE.ZIP). I downloaded all of these files, unzipped them, and put them in a directory. All of the programs come with documentation, but the average user will only need to read the documenation for QCD-EZE as Duncan documents how to install the ATAPI and CDRM drivers.

Briefly, here is what I added to my BOOT program:

```
LRESPR ATATIQ40_BIN
LRESPR CDRM_BIN
CDR_DRIVE 1,1,1
```

The LRESPR's load the two drivers. The CDR_DRIVE command tells the drivers about the local CD-ROM. The first 1, is for the device number. With the drivers loaded, I now have a CDR device. Since I only have 1 CD-ROM, I am calling it **CDR1_**. If I had used a 2 for the first number in the CDR_DRIVE command, I would have a **CDR2_** device. The second number is the physical device number. On my Q40, I only have 1 IDE controller and the CD-ROM is the second device on the controller. In physical terms, device 1 is called **0**, device 2 is called **1**. If I had a second IDE controller, then the first device on that controller would be called **3**, the second device **4**, and so on. The last number in the CDR_DRIVE command is the session number. I really don't know what this means, so I followed the documentation that Duncan write and used Session 1. After I made these changes to my BOOT, I saved it and rebooted the Q40. Once I rebooted, I typed in CD_EJECT and the CD-ROM door popped open. So, I know that part was working.

I decided that the best CD to experiment with would be the QL Today cover CD-ROM that was sent out a few months ago. I put it in the tray and typed CD_LOAD. The tray moved back into the CD-ROM drive and I could hear it start spinning. I then executed QCD-EZE. It took me a minute to figure out that QCD-EZE needs to be told where the CD is located. Since I have the CD located at CDR1_, I had to excute QCD-EZE like this:

```
EXEC QCDEZE_OBJ;"CDR1_"
```

After a few seconds, a window popped up with toolbar and a blank area. In the blank area was a few folder looking icons. These folders are the directories on the CD-ROM. I could click on the folders to open a directory. I could click on a **back** icon to close that folder and return to the higher directory.

The QL Today Cover CD has both ISO 9660 directories and a QXL.WIN file. QCD-EZE would allow me to browse both the ISO 9660 directories and the QXL.WIN file. Obviously, if I saw an executable in the ISO 9660 directories I would not want to copy it to my Q40 since all the header information would be gone. But, from the QXL.WIN file, all the header information is intact and I could easily copy executables from the CD-ROM.

QCD-EZE, besides the CD-ROM drivers, also requires ToolKit II and QMENU. It knows how to use FileInfo, so you will need to have FileInfo installed if you wish to utilize that part of QCD-EZE.

The Toolbar in QCD-EZE contains icons for the following commands:

- Move
- FileInfo II Execute File
- Backup one Directory Level
- Copy File(s) to Another Drive
- Copy File(s) to the Scrap
- Copy File Name to the Stuffer Buffer
- Change the Size of QCD-EZE's Window
- CD Eject/Load
- Quit

If you right click on an Icon in the Files window, the file is selected. If you left click on an Icon, a window pops up with information about that file. If you want to select more than one file, you can hit the TAB key. This will toggle QCD-EZE in to multi-selection mode.

I really have not used the CD-ROM for much than just "gee-wiz", but I hope to in the future. Using FileInfo II, I can configure QCD-EZE to view JPEG images that I've burned onto a CD. I can use it to transfer larger files from the PC to the Q40 by burning CD's.

In summary, the installation of the drivers is easy. The installation of QCD-EZE is also very simple, and the quality of the QCD-EZE rivals commercial PE programs. For those with more interest in the ISO 9660 format, the source code comes with QCD-EZE.

Quanta Workshop and AGM 2004

Dilwyn Jones

A Quanta workshop and 2004 AGM was held at the 3rd Davyhulme Scout Headquarters in Davyhulme, Manchester on the 17th and 18th April 2004. This is an established and ideally located venue, being just the right size and close to major roads etc (just off the M60 motorway), with good facilities including a small catering area to keep keen QLers fed and watered.

Attendance could not exactly be described as good, since the total number of visitors over both days struggled to

get over about 50. Nonetheless, it was an enjoyable event with plenty to do and see.

As usual, there was a wide variety of systems there, including Auroras, PC/QPC systems and Q60, but increasingly there seems to be fewer and fewer original Sinclair-cased black QLs. We have come a long way during the QL's lifetime! As so many "QL" systems are now boxed into similar cases to Q60s and PCs, it can be an interesting experience trying to figure out what type of system you are looking at.

The close proximity of the date of the North American QL show (the following weekend) meant that some of the traders were not present because they were flying out early for the USA show. Traders present included Q-Branch (Roy Wood), D&D Systems (Derek Stewart), and Geoff Wicks.

Q-Celt Computing did not attend due to family commitments. TF Services (Tony Firshman) and Jochen Merz did not attend because of the USA show, while Bill Richardson and RWAP Software (Rich Mellor) were also absent. Rich Mellor still suffers from ill health, but is very active behind the scenes - for example he has managed

JOCHEN MERZ SOFTWARE

Im stillen Winkel 12 D-47169 Duisburg
Tel. 0203 502011 Fax 0203 502012
http://smsq.j-m-s.com smsq@j m s.com

text87plus4 PATCH!
You can order it NOW!

**Now for
Q40, Q60,
Aurora!**

text87, one of the few programs which does not run properly in high-colour mode can now be modified to work in 16 bit colours on the QXL, QPC, Q40, Q60 and Aurora (with SMSQ/E Version 3). Marcel has added a new driver for this program which will be added to the program and enable you to run it nicely in high-colour mode. You have two options: it can look the same way as before (white/red/green on black) or - much better looking - colours on a nice grey. It is up to you, you can try both settings.

**text87plus4 PATCH is available now from
J-M-S for only EUR 12,- (incl. p&p!)**

Easy-to-use, will it provide you with a text87 which works without having to re-start QPC in a different colour mode or use a DISP_COLOUR 0 which will screw up the rest of the display. No, with text87plus4 PATCH it will work without the need to change display resolution - and we think it looks even better in grey!

Please note: the PATCH program requires you to own the latest version of text87plus4 English (E4), file size 116850 bytes or the latest version of text87plus4 German (G4), file size 117354.

You do not need to configure or modify the result of using text87plus4 PATCH - just execute this one instead of your original program!

The Patch is sold with the permission of text87 author, Fred Toussi.

You will find a secure contact form on the J-M-S homepage now, where you can submit credit card data etc. without having to send it via email:

SMSQ.J-M-S.COM

to raise funds for Quanta by selling donated equipment on behalf of Quanta.



Chris Grogan tries to figure out whether he's looking at a PC or a QL

On Saturday afternoon, Roy Wood gave an interesting talk on the current situation regarding hardware products from Nasta (Zeljko Nastasic of Croatia). Following a spell working in America, Nasta has now returned to Croatia to work. His enthusiasm for QL projects continues, but has found it difficult to devote the time he would like to existing and proposed projects. For example, there are parts available to produce more of certain products if deemed worthwhile and the finance is available. Roy discussed Aurora, Goldfire, Qubide 2 (or whatever it will be called) and the possible Ethernet card. Some discussion also took place on the possibilities (or not) of a USB interface. With some of the projects, software design may be a major issue - we all remember I'm sure how long it took for release of "colour drivers" for the Aurora. It showed how important this topic was since all activity in the main hall stopped as everyone drew up chairs around Roy as he gave the talk.

I gave an ongoing demonstration at my table of the Launchpad GUI and was often asked about QDT, the other GUI soon to be released by author Jim Hunkins. Jim has made no secret that QDT will be a high end

product, probably requiring SMSQ/E and GD2 systems, and hopefully demonstrated at the USA QL show. In the past, we have often shied away from GUI (Graphical User Interface) systems on the QL, preferring command line operation or relatively simple task switcher programs like Taskmaster, but with the increasing acceptance of what is going on with other computer systems outside the QL scene, there is also a realisation that it is undoubtedly the way forward for most users. Certainly, properly implemented and used, they can make our computing life easier.



Workshop organiser Sarah Gilpin

Geoff Wicks launched a new program called Pin-Down - you may have seen the teasers in recent Just Words adverts about this program. This program is best described as a fun way to remember PIN codes, those little security numbers you have to remember to withdraw cash, unlock something or gain access to your computer. This program is a help to memorising those numbers - it generates rhyming phrases to help you remember sequences of numbers. An interesting example of a completely new piece of QL software.

One rather nice aspect of this workshop was that a few ex-QLers turned up to have a look what was going on with

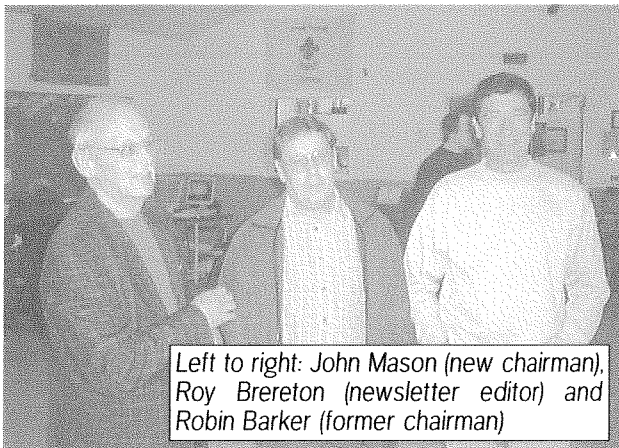
the QL scene these days, and to look up old friends from the QL scene. It is heartening that such people are finding out about these events (presumably from the internet or still being in contact with QLers) and attending. I gave a copy of my emulators CD to these people in the hope of tempting them back to the QL scene!

One of these was David Batty, head of former QL software house Sector Software. David still lives at the premises from which Sector Software traded in QL days, and one of his QL products Typing Tutor is still going strong as a PC product. One interesting snippet I learned from David was that former QL software house Digital Precision has been officially closed, but the name has now been registered by David Batty at companies house - in effect one former QL software house also now owns the name 'Digital Precision'!

The Saturday workshop was set up as an afternoon-only event, to allow visitors from further afield to travel on the Saturday morning.

The meeting carried on slightly later than anticipated, and was followed by a dinner at the Pond Quay, attended by a couple of dozen people.

The Sunday started quietly, although more people came in the afternoon in time for the Annual General Meeting. Roy Brereton started the AGM slightly earlier than planned in his usual efficient manner. The AGM took place in the main workshop room rather than in a separate room as has been past practice, which was a good move as it ensured everyone present sat down to attend the AGM rather than some staying at their tables and missing the AGM.



Left to right: John Mason (new chairman), Roy Brereton (newsletter editor) and Robin Barker (former chairman)

Chairman Robin Barker opened the meeting and explained his reasons for stepping down as Chairman. His work has meant being out of the country more and more of late and as a result he has found it increasingly difficult to devote enough time to his duties as Chairman. Robin is to be thanked for his years of service to the QL scene, from way back in time to his days as QL trader Di-Ren (did you know that the name originally stood for Disco-Rentals!) and his years on the Quanta committee. John Mason now takes over as Chairman - John has been a Quanta official for some time and is sure to bring his wealth of experience to the new duties.

Roy Brereton explained that he had taken over as acting editor of the newsletter, following the resignation of Paul Merdinian. Roy explained he was actually enjoying the job and appealed for contributions for the newsletter - please send all articles or letters for the newsletter to Roy Brereton for the time being.

At the previous AGM, the possibility of the newsletter going electronic had been discussed (e.g. emailed to members). At the time, it was felt this might be the way forward as it was feared that future income and costs might not sustain the current model of operation.

However, cost savings etc had meant that by this year Quanta's financial situation was much better than had been feared and this coupled with the fact that fewer members than thought had access to

email etc had in effect meant that the proposal to go electronic was a non-starter, at least for the time being.

John Gilpin explained the situation regarding insurance. There had been some changes resulting in the fact that Quanta may no longer be able to insure sub-group meetings, although this may not be as much of a problem as initially thought, as meetings were usually held at venues which had their own public liability insurance - discussion on this issue was suggested. Quanta itself and the workshops continue to be insured.

Geoff Wicks gave some information regarding the QL 2004 event in The Netherlands later this year.

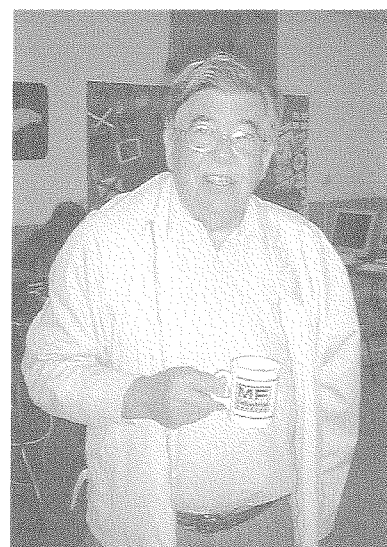
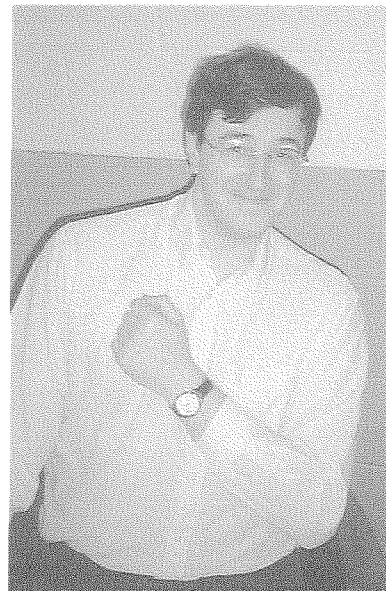
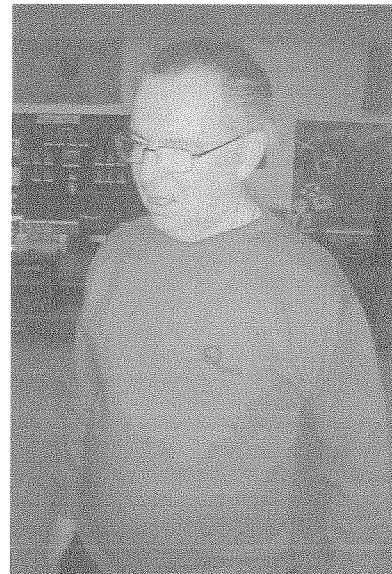
John Gregory appealed for new contributions to the software library, which had fallen somewhat of late.

The new committee was voted in largely through the fact that the number of proxy votes was greater than the number of members present at the AGM. The new committee looks like this:

Chairman - John Mason
 Secretary - John Southern
 Treasurer - John Gilpin
 Committee members - Roy Brereton, John Gregory and Geoff Wicks.

Roy Brereton assumes the mantle of newsletter editor,

John Gregory remains as software editor and John Gilpin keeps his role as head librarian.



Top to bottom: John Southern, John Gilpin, John Gregory

What are blobs and patterns?

Wolfgang Lenerz

I've had some questions recently on blobs and patterns, what are they used for, how can one use them etc....

To cut a long story short, the blob is a sort of see-through mask through which colours may be dropped onto the screen. It is like a sieve, with one hole for each of the pixels covered by the blob.

This mask can either let a colour drop onto the screen, or not. Thus, each hole in the sieve lets the colour fall onto the underlying pixel or it doesn't. If it lets the colour drop, then the underlying pixel now has the colour that dropped through this mask. If not, then the screen will continue to show the old background colour it already had before the blob (and pattern) were applied to it.

1 – How to make a blob from S*Basic with QPTR.

You make a blob in the same way you would make a sprite. The thing to remember is that only the "mask" information will be taken into account. A pixel marked as transparent (denoted with a space) will let the background colour through, any "coloured" pixel will be treated as non transparent. In the example program below, the blob is made as follows:

```
blob1 = ALCHP(SPRSP(20,20))
      : REMark blob1 (mask)
SPHDR blob1,20,20,0,0,4
      : rem make header
linnum%=0
FOR lp%=1 TO 20
  a$=FILL$("r",20-lp%)&FILL$(' ',lp%)
  SPLIN blob1,linnum%,a$
END FOR lp%
```

The line `a$=FILL$("r",20-lp%)&FILL$(' ',lp%)` makes a string 20 chars long, filled at first with the letter "r" repeated 20 times and then this is progressively filled with spaces which replace the "r"s.

When the blob will be applied to the screen with a pattern, the effect will be a block of colour slowly diminishing to the left, as each line of the blob lets more and more colour from the background shine through.

Since blobs are just masks through which you drop a colour onto the screen, a blob in itself has nothing visible. To make its effects visible, you need a pattern to provide the colour information

which is then dropped onto the screen through the blob.

2 – And Patterns?

Patterns are the contrary to blobs. They just contain colour information: one colour per pixel.

This colour is then dropped onto the screen through the sieve of the blob. If, for any pixel, the blob is not transparent, then the colour of the pattern for that pixel will be applied to that pixel. If the blob is transparent, the background colour, not the pattern colour, will be shown.

Of course, even the pattern can be "transparent", in which case it has no colour information and the background pixel colour will shine through.

Patterns are made just like sprites and blobs. Here is a small example program that may show you the effect of blobs and patterns.

```
100 DEFine PROCEDURE init
110 LOCAL lp%,linum%
120 blob1 = ALCHP(SPRSP(20,20))
      : REMark blob1 (mask)
130 SPHDR blob1,20,20,0,0,4
140 linum%=0
150 FOR lp%=1 TO 20
160   a$=FILL$("r",20-lp%)&FILL$(' ',lp%)
170   SPLIN blob1,linum%,a$
180 END FOR lp%
190 :
200 blob2 = ALCHP(SPRSP(20,20))
      : REMark blob2
210 SPHDR blob2,20,20,0,0,4
220 linum%=0
230 FOR lp%=1 TO 20
240   a$=FILL$("r",lp%)&FILL$(' ',20-lp%)
250   SPLIN blob2,linum%,a$
260 END FOR lp%
270 :
280 patt1 = ALCHP(SPRSP(20,20))
290 SPHDR patt1,20,20,0,0,4
      : REMark pattern (colours)
300 linum%=0
310 FOR lp%=1 TO 20
320   SPLIN patt1,linum%,'aaaaaaaaaaaaaaaaaaaaa'
330 END FOR lp%
340 :
350 patt2 = ALCHP(SPRSP(20,20))
      : REMark another pattern
360 SPHDR patt2,20,20,0,0,4
370 linum%=0
380 FOR lp%=1 TO 20
390   SPLIN patt2,linum%,'grgrgrgrgrgrgrgrgr'
400 END FOR lp%
410 END DEFine init
420 :
430 DEFine PROCEDURE p
440   init
450   CLS#2
460   WBLOB#2,20,20,blob1,patt1
470   WBLOB#2,20,50,blob1,patt2
480   WBLOB#2,20,80,blob2,patt1
490   WBLOB#2,20,110,blob2,patt2
500 END DEFine p
510 :
```

Procedure P writes the blobs and patterns to the screen. Play around with the different colours and you'll see exactly how these two concepts work together.

3 – What about extended colours?

You can also make extended colour blobs and patterns. Here, however, the QPTR keywords are of not much help, you will need to POKE the values right into memory. Indeed, conceptually extended colour blobs and extended colour patterns are the same as mode 4 blobs and pattern, it's just the colour information that changes.

We'll take example on mode 64 (24 bit colour) blobs and patterns.

4 – Blobs

You can set up a 20x20 blob, in mode 64 (24 bits) with the following program:

```
xs%=20:ys%=20 : REMark x, y sizes
header=HEX("22") : REMark size of header
size=xs*ys*4*2+header : REMark size of blob
blob1=ALCHP(size) : REMark get the memory needed
POKE_W blob1,HEX('0240') : REMark form 1 -
                        show a mode64 blob
POKE_W blob1+2,0 : REMark form 2 (not applicable)
POKE_W blob1+4,xs% : REMark x size
POKE_W blob1+6,ys%
patmask=blob1+header : REMark where pattern mask
                        goes
ptr=blob1+16 : REMark pointer to pat mask goes
POKE_L ptr,patmask-ptr : REMark make pointer
col=-1 : REMark a totally opaque RGB mask
        : REMark (last byte ignored)
FOR ylp%=0 TO 19
  FOR xlp%=0 TO 19
    POKE_L patmask,col : REMark poke the mask
    patmask=patmask+4
  END FOR xlp%
END FOR ylp%
```

As you can see, the mask is simply poked into memory. This is set to -1, which is \$FFFFFFF in Hex. This means that all pixels will show the pattern colours.

Things get a bit more interesting and, unfortunately, complicated here. In mode 64, each pixel is represented by 8 bits (one byte) per colour: one byte for the Red component, one for the Green component and one for Blue. To make up a full long word (= 4 bytes), a 0 byte is appended after these, hence one pixel is represented by R G B 0. The blob represents a mask allowing one pixel to show the background colour (transparent) or the pattern colour (opaque). It is recommended that, if any pixel is supposed to be visible in the

pattern colour (opaque blob), then the three corresponding bytes should be set to -1 (\$ff). If a transparent pixel is wished, the three bytes should be set to 0, which will make the pixel show the background colour (transparent).

You may also set the value for each of the three colours to some intermediary value, but in that case you may get some unforeseen results...

Also note that blobs need no colour information, hence this is not even poked into memory.

5 – Patterns

For patterns, this is much of a sameness. Let's look at the following example:

```
patt1=ALCHP(size)
POKE_W patt1,HEX('0240') : REMark form 1
POKE_W patt1+2,0 : REMark form 2 (non applicable)
POKE_W patt1+4,xs% : REMark x size
POKE_W patt1+6,ys%
patmask=patt1+header : REMark where pattern mask
                        goes
ptr=patt1+12 : REMark pointer to pat mask goes
POKE_L ptr,patmask-ptr : REMark make pointer
col=HEX('FFFFFFF') : REMark colour - white !!
FOR ylp%=0 TO 19
  FOR xlp%=0 TO 19
    POKE_L patmask,col
    patmask=patmask+4
  END FOR xlp%
END FOR ylp%
```

Again, the necessary values are poked into memory. The colour information goes into the necessary space pointed to by the required pointer. The colours are made up of the individual RGB bytes, as mentioned above.

Unfortunately, colour patterns may also have masks (whatever for?). If you do not set the mask, as in the above example (only the colour information is given), there is no problem whatsoever. You may, however, also specify a mask. You would do this by adding the following to the above example:

```
ptr=patt1+16 : REMark pointer to pat mask
POKE_L ptr,patmask-ptr : REMark make pointer
col=HEX('FFFFFFF') : REMark mask -> always -1
FOR ylp%=0 TO 19
  FOR xlp%=0 TO 19
    POKE_L patmask,col
    patmask=patmask+4
  END FOR xlp%
END FOR ylp%
```

It is, again, strongly recommended to leave the masks for the patterns at \$ff for each colour, i.e. the whole pattern mask should be -1. Here again, you may use other values, but then you will get unforeseen colours on the screen.

Letter-Box

Simon Goodwin writes:

QL & 68000 User Group

This vintage club has now moved back to its previous venue, the Queen's Head pub in the City Centre. The Queen's Head is in Steelhouse Lane, central Birmingham, four minutes walk from Snow Hill main-line railway station, and has ample street parking nearby.

Meetings are usually on the first and third Monday of each month (but not on bank holidays unless the previous one was cancelled for that reason). After their first free meeting attendees pay £1 each time towards the organisation and monthly newsletter costs – except once a year when the group pays all those who attend a pound instead 8–)

We provide food twice a year for those who attend the AGM and the group's birthday party. At meetings we discuss all sorts of things, QL-related and otherwise, usually from about 8:20pm onwards till 10:30 or later. The club owns QL hardware, a disk library and subscriptions to relevant magazines (guess which). People can keep in touch by joining the mailing list for £3.50 per year, even if they can't attend regularly.

Club Secretary Mike Bedford-White can be contacted by telephone on 0121 708 2560, anytime after 11am.

Plain text email enquiries are welcomed by me, Simon N Goodwin, via qdos@studio.co.uk

Simon Balderson writes:

Dear editor,

Just a few comments to add to Roy Woods comments re. Bytes & Pieces in QL Today Vol. 8 issue 6.

I think a possible way forward to help solve the perennial problem of printer drivers for the QL is with the ProForma driver included with ProWess. ProForma is able to print to modern Graphical User Interface printers which treat a screen full of text as though it were a picture of a printed page (basically printing part of video memory), rather than translating a stream of ASCII codes into text as with the older printers with built in founts. ProForma comes with several drivers for different brands of printer and in recent years RWAP Services have taken the time and effort to develop

and much improve the Epson ESC/P2 drivers.

Seeing as ProWess is now free it would seem a good idea to push ProWess as a solution to the QL's printer compatibility problems. After all, the bulk of the work is already done in ProForma allowing the QL to print raster founts and graphics. ProForma does need updating as it is very limited in its graphics capabilities ie. it can not handle JPEG images, which is probably the most common image format around now. It can print GIF images but is painfully slow when displaying them on screen and on paper the quality isn't always too good. As far as I know there has been no further development of ProForma since RWAP Services upgraded the Epson drivers.

If a decision was made in the QL community to adapt ProWess and ProForma as the official printer driver then perhaps the suggestion that Geoff Wicks has put forward that funds QUANTA holds could be used as an incentive to further develop it. This could include, perhaps, adding utilities or filters to convert the output of older software like the PSION suite, and programs that were designed for use with dot matrix printers back in the late 1980's, to raster graphics format. It would not be a perfect solution but without the army of professional programmers that the PC world has to write software for every new printer coming on to the market, we do not have much choice. Tell me if I'm wrong in thinking that ProWess works in all versions of SuperBasic and SMSQ/E. If it does, then it will allow printing on all QL platforms, be it the original black box, a Q40 or QPC2. Some decision will have to be made soon as printers with built-in founts will probably disappear altogether as manufacturers who only see through Microsoft shaped Windows continue to cut costs.

George Gwilt writes:

Letter to QL Today – 6th April 2004

In my article Turbo and Parameters in QL Today Vol 8 Issue 5 Jan/Feb 2004 there is an assembler program EXT1.asm starting on Page 12.

Unfortunately in the printing the indentations were all left out so that anyone trying to type it and assemble it as it stands would find a very large number of errors reported. Since it is intended as a template for machine procedures and functions which can take array parameters or change the value of parameters and yet be called in code compiled by Turbo it is important for those using the program that it be correct.

[Very sorry about that - not that only I did not notice, it also fell through the proof-reading- Editor]

The correct version is given below:

```

; EXT1_asm
; To show how Turbo can deal with
;     (a) Altered parameters
;     (b) Array parameters
;
; This sample procedure EXT1 a%,a$,v%
; sets a%=v%(a%) where v% is a one dimension integer array
; and strips off a trailing _asm (regardless of case) from a$
;

BV_VVBAS EQU    $28
BV_RIP   EQU    $58

START    LEA     DEFINE,A1
         MOVE.W  BP_INIT,A2
         JMP     (A2)

DEFINE   DC.W    1    one proc
         DC.W    EXT1-*
         DC.B    4,"EXT1"
         DC.W    0    end of procedures
         DC.W    0    no of functions
         DC.W    0    end of functions

EXT1     LEA     24(A3),A0
         CMPA.L  A0,A5
         BNE    BAD_PAR    ---->    3 parameters needed
         BSR    TURBT      are we in a Turbo program? . .
         BNE    NON_T      . . no - act normally
;
; Here we expect pointers to Turbo's Vector Table in place of the
; parameters
;
         MOVEA.W CA_GTLIN,A2
         JSR    (A2)
         BNE    BAD_PAR    ---->
;
; Now pointers to the Vector Table are on the Maths Stack
;
         MOVEA.L (A1,A6.L),A2
         MOVEA.L (A2),A2          pointer to the value of a%
         MOVE.W (A2),D2          a% is now in D2.W
         MOVEA.L 8(A1,A6.L),A0    Vector Table entry for v%
         MOVEA.L (A0),A0          pointer to the descriptor
         TST.W   8(A0)            One dimension? . .
         BNE    BAD_PAR    ---->    . . no
         CMP.W   10(A0),D2        In range? . .
         BGT    0_RANGE    ---->    . . no
         MOVEA.L (A0),A0          pointer to values
         ADD.W   D2,D2
         MOVE.W (A0,D2.W),(A2)    set the value in a%
;
; Now the string
;
         MOVEA.L 4(A1,A6.L),A0
         MOVEA.L (A0),A0          pointer to the descriptor
         CMPI.W #-1,8(A0)        no dimensions? . .
         BNE    BAD_PAR    ---->    . . no - not a simple string
         MOVEA.L (A0),A1          pointer to string
         SUBA.L  A6,A1            pretend to be in Basic
         BSR    STR              deal with the string
END     MOVEQ   #0,D0
        RTS
;
; Here is the "normal" processing
;
NON_T   MOVEA.W CA_GTINT,A2
        LEA    8(A3),A5          for 1 parameter
        JSR    (A2)
        BNE    BAD_PAR    ---->
        MOVE.W (A1,A6.L),D2      a% to D2.W

```

```

MOVE.W    16(A3,A6.L),D0      Name Table entry for v%
ANDI.W    #$FF0F,D0
CMPI.W    #$303,D0           an integer array? . .
BNE       BAD_PAR    -----> . . no
MOVEA.L   20(A3,A6.L),A2     pointer to descriptor rel to VV_BAS
MOVE.L    BV_VVBAS(A6),D0    relative start of VV_BAS
ADDA.L    D0,A2              -> descriptor rel to A6
CMPI.W    #1,4(A2,A6.L)     one dimension? . .
BNE       BAD_PAR    -----> . . no
CMP.W     6(A2,A6.L),D2     In range? . .
BGT       O_RANGE    -----> . . no
MOVEA.L   (A2,A6.L),A2      -> itmes rel to VV_BAS
ADDA.L    D0,A2              -> items rel to A6
ADD.W     D2,D2
ADDA.W    D2,A2              -> v%(a%) rel to A6
MOVE.W    (A2,A6.L),(A1,A6.L) set a%
MOVE.L    A1,BV_RIP(A6)
MOVEA.W   BP_LET,A2
JSR       (A2)
BNE       BAD_PAR    ----->

;
; Now the string
;
MOVEA.L   A5,A3
LEA       8(A5),A5
MOVEA.W   CA_GTSTR,A2
JSR       (A2)
BNE       BAD_PAR    ----->
BSR       STR
BNE       END          no change in string
MOVE.L    A1,BV_RIP(A6)
MOVEA.W   BP_LET,A2
JSR       (A2)
BRA       END

;
BAD_PAR   MOVEQ    #-15,D0

O_RANGE   RTS
MOVEQ     #-4,D0
RTS

; Subroutines
;
; TURBT sets CC Z if compiled by Turbo
; Uses no registers
;
TURBT     MOVEM.L  A0-1/D0-2,-(A7)
MOVEQ     #-1,D1          This Job
MOVEQ     #0,D2          Top Job
MOVEQ     #MT_JINF,D0
TRAP     #1
CMPI.L    #"Turb",34(A0) EQ if compiled by Turbo
MOVEM.L   (A7)+,A0-1/D0-2
RTS

;
; STR does the work needed to see if the string needs shortening.
; (A1,A6.L) points to the string.
;
STR       MOVE.W   (A1,A6.L),D0      length
SUBQ.W    #4,D0                    set reduced length
BMI       STEND                    too short - can't be _asm
LEA       2(A1,D0.W),A2            point to last 4 characters
MOVEQ     #3,D5                    count - 1
ST1       ROL.L    #8,D3            move up a byte
MOVE.B    (A2,A6.L),D3            put in the next byte
ADDQ.L    #1,A2                    step to the next character
DBF       D5,ST1                  count to four characters
ANDI.L    #$FFDFDFDF,D3          set 'asm' to upper case
CMPI.L    #'_ASM',D3            Was it "_ASM"? . .
BNE       STEND                    . . no
MOVE.W    D0,(A1,A6.L)          set the reduced length
MOVEQ     #0,D0                  mark a change occurred

STEND     RTS

```

TF Services

Compswitch

A UK 4-way trailing socket designed to switch off computer peripherals automatically when the computer is switched off, or (in the case of an ATX computer) when it auto-powers down. *Compswitch* has one control socket, and three switched sockets. Can be used with lights/hifi/monitors—ie a QL monitor can be used as a switch control.

Cost **£24**

superHermes

A major hardware upgrade for the QL
All Hermes features (working ser1/2 at 19200, independent baud rates/de-bounced keyboard/keyclick) IBM AT kbd I/F // HIGH SPEED RS232 at 57600// serial mouse port and 2 other RS232 inputs// 3 I/O lines // EEPROM

Cost (including manual/software) **£90** (£92/£93)
IBM AT UK layout Keyboard..... **£11** (£13/£15)
Serial mouse..... **£8** (£8.50/£9)
Capslock/scrolllock LED **£1** (£1.50/£1.50)
Keyboard or mouse lead **£3** (£3.50/£3.50)
High speed serial (ser3) lead..... **£4** (£4.50/£4.50)

Hermes available for **£25** (£26/£27) Working ser1/2 and independent input, debounced keyboard.

SuperHermes LITE: All Hermes features (see above) + an IBM AT keyboard interface only.

Cost (incl keyboard lead) **£53** (£54/£55)

QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

£27 incl 6 month guarantee

Minerva

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

First upgrade free. Otherwise send **£3** (+£5 for manual if reqd).

Send disk plus SAE or two IRCs

MKI...**£40** (£41/£43) MKII...**£65** (£66/£67)

MINERVA RTC (MKII) + battery for 256 bytes ram.
CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

QL RomDisq

Up to 8 mbyte of flash memory for the QL
A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card, and reads at some 2 mbytes per second. Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off RomDisq at hard disk speed with only a memory expansion needed.

2 mbytes RomDisq..... **£39** (£40/£41)
4mbytes RomDisq..... **£65** (£66/£67)
8 mbytes RomDisq..... **£98** (£99/£100)
Aurora adaptor..... **£3** (£3.50/£4)

MPLANE

A low profile powered backplane with ROM port

A three expansion backplane with ROM port included for RomDisq etc. Aurora can be fitted in notebook case and powered off single 5V rail - contact QBranch for details. Two boards (eg Aurora and Gold Card/Super Gold Card/Goldfire fixed to base. Suitable for Aurora (ROM accessible from outside) & QL motherboard in tower case. Specify ROM facing IN towards boards, or OUT towards back of case.

Cost **£34** (£35/£36)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface 16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)

2 amp (for 8 relays, small motors) **£40** (£43/£44)

4 amp total (for motors etc) **£45** (£48/£50)

Relays (8 3a 12v 2-way mains relays (needs 2a power driver) **£25** (£28/£29)

Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required..... **£25** (£27/£28)

Analogue Interface Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temp measurements, sound sampling (to 5 KHz), x/y plotting..... **£30** (£31/£32)

Temp probe (-40°C to +125°C)..... **£10** (£10.50/£11)

Connector for four temp probes..... **£10** (£10.50/£11)

Data sheets..... **£2** (£2.50/£3)

Control software & manual (for all I/F)..... **£2** (£2.50/£3)

QL SPARES

Keyboard membrane no longer on sale
1377 PAL **£3** (£3.50/£4)
Circuit diagrams **£3** (£3.50/£4)
68008 cpu or 8049 IPC..... **£8** (£8.50/£9)
8301/8302 or JM ROM or serial lead **£10** (£10.50/£11)
Power supply (sea mail overseas)..... **£12** (£19/£23)
Other components (sockets etc) also available

Prices include postage and packing (Airmail where applicable) Prices are: UK (Europe/Rest of world). Payment by cheque drawn on bank with UK address/postal order or CASH! I can no longer accept card payments as UK only does PDQ transaction. SAE or IRC for full list and details 22 Feb 03

29 Longfield Road, TRING, Herts, HP23 4DG

Tel: +44 (0) 1442-828254

Fax/BBS: +44 (0) 1442-828255

tony@firshman.co.uk

http://www.firshman.co.uk

Some Solutions to the Dreaded Printing Problem

John Gregory

The story starts when I was asked recently to assist a shopkeeper who was using a real QL in its black box to run the stock control side of his business. This he had done for nearly twenty years using a Superbasic program he had had written for him and which exactly suited his needs. He now realised that his hardware was obsolescent and could fail at any time leaving his business vulnerable. However he wanted to retain his program with which he was so familiar. The obvious answer was to use QPC2 running on the PC he already had for other uses. There were some minor problems in converting for SMSQ/E a program which had been written in the early days when programmers often wrote tortuous undocumented code. These problems were solved but the more interesting feature of the job was how to tackle printing, a key part of the operation to satisfy the VAT inspector. The original QL was sending its output to an ancient dot-matrix printer which could also fail without warning. The PC is fitted with a Windows printer connected to a USB port which readily can be changed for a new one if it fails, so how to use it for printing rather than a dot-matrix connected to LPT1 was the question. QPC2 provides the link between the SMSQ/E environment and Windows in the form of the DOS device. I altered the SBASIC program to print all its output to a file starting with the line `OPEN_OVER #3, "DOS7_OUTPUT.TXT"` rather than `OPEN #3, SER1`

`OPEN_OVER` avoids the need for the user to answer an overwrite prompt. So now the "QL" output was available to Windows programs. When you are writing a program for someone else you want its operation to be as smooth and simple as possible. This was where another feature of QPC2 provided the perfect solution: the `QPC_EXEC` command. This will start a Windows program directly from QPC2. In this case I used

```
QPC_EXEC "WORDPAD",  
"I:\OUTPUT.TXT"
```

where drive I:\ is configured in QPC2 to be DOS7... Physically it was a Compact Flash card! How to pass a file name to a Windows program is documented in the QPC2 manual but I realised that it is also possible to indicate to Wordpad what to do with the file having loaded it. Adding the parameters `/p %1` after the file name will cause Wordpad (and Word) to start printing straightaway. The reasons for using Wordpad in these examples are that it is available on all Windows systems, loads quickly and is simple to use. The final line of the SBASIC program was `QPC_EXIT` which closes QPC2 down neatly and leaves just the Windows environment. That completes the chain: direct printing of text from SBASIC to a Windows printer.

The next question was how to incorporate text variations such as bold, underlined etc. Any occurrences of Epson codes in the SBASIC source such as `CHR$(15)` to give compressed print will result in a little square

being printed by Wordpad. Not much use. If only a little of the text needs modification then it can be done manually in Wordpad itself. A more comprehensive answer is to employ HTML or Rich Text Format tags. These are in plain text and can be added to the `PRINT` statements in the SBASIC program. Examples are `<i>` in HTML or `\i` for RTF to turn on italics. These will be interpreted by the appropriate Windows program when run: e.g.

```
QPC_EXEC "IEXPLORE",  
"I:\OUTPUT.HTM"
```

or

```
QPC_EXEC "WORDPAD",  
"I:\OUTPUT.RTF"
```

Roy Wood gives a useful list of the more common HTML tags with his program 'The HTML Machine' which is available from the Quanta Library! I found out about RTF tags by incorporating the effects I wanted in a file which I saved from Wordpad in RTF format. Reading the file in a text editor which could not interpret them revealed what they were.

All this led me to consider how to extend the idea to other areas where there would be no practical possibility of altering the program itself, such as the Psion programs which still have quite a following. Xchange is much more flexible than the stand-alone programs so giving the best starting point. Its Quill has an export command supporting printing to a file. However it refuses to accept a file name of the form `DOS7_OUTPUT.TXT`

Then I remembered a tip by Stuart Honeyball for printing to network printers. Xchange Quill will not accept `n2_par` but will accept `_n2_par`. The same is true with DOS filenames:

```
_DOS7_OUTPUT.TXT  
works!
```


The required file is created less the initial "_". A prompt to overwrite the file is given whether it exists or not. The same trick works with all the Quill, Abacus and Archive export commands and with Spoolon in Archive. The Quill and Abacus print commands also work. If no printer_dat file is set up then no unprintable characters will be passed to the DOS files. Printing in the Windows environment can then be carried out as from SBASIC programs. The only Psion program from which printing is not possible is Easel. It should be possible to handle text bold, italics etc. by

including the necessary tags in the printer_dat file but there's more work to be done to prove this. How can you write an article about printing without mentioning printer_dat files!

Finally in the field of text preparation there are Perfection and Text87. My knowledge of the former is sparse and Text 87 has a list of acceptable devices embedded in it and DOS is not included, along with ROM. One could export to an acceptable device such as RAM1_ and set up a button to copy the file to a DOS device. An alternative is to patch Text87 to replace the unused MDV device by DOS;

Masterspy is ideal for this job. Text87 can create files of the form OUTPUT.TXT.

What about graphics? PC users will be aware of using the Print Screen key to copy the screen to the Clipboard which can then be pasted into Word for printing. Less well known is the feature of accompanying this with the ALT key which will copy only the currently active window.

I realise that none of this is of any help to the users of native QL hardware (of which I am one) but it should be of use to owners of QPC2.



Caption Competition

Dilwyn Jones

Our caption competition photo was taken at the USA QL show and shows Phoebus Dokos (left), an electric drill and Jim Hunkins (author of QDT).

My immediate reaction was that Phoebus was attempting to gain access to the QDT source code by drilling direct into Jim's brain, but you might have a rather better caption than that!

I am offering a free copy of Launchpad to the sender of what Jochen and I consider to be the best caption for this photograph. If the sender turns out to already be a Launchpad user, I'll offer any one of a range of QL CDs instead.

Send all entries to Jochen or myself at the addresses shown inside the front cover of this issue.

Replies accepted until Sat., 10th of June, 2004.

Disassembling Register

Lists

George Gwilt

In QLtdis part 11 published in QL Today Vol 8 Issue 5, Norman Dunbar describes how he disassembles register lists. I recollected that some time ago I performed the same operation for NET_PEEK and GWDISS, so I compared the two methods. I found that we were essentially producing the same output.

There were two differences, the second large enough to warrant this note.

The first difference was in presentation. I had decided to use the standard separator "slash" between registers. Norman uses "comma". Although my output for register lists can be assembled as it stands unlike Norman's, this in no way should be grounds for criticism since in general my disassembled output cannot be re-assembled as it stands.

Where Norman would produce:

```
MOVEM.L D0,D4-6,--(A7)
```

I would have:

```
MOVEM.L D0/D4-6,--(A7)
```

The second difference is in the method used. Norman uses a table; I use code.

Norman's table contains the required register strings for every possible array of register numbers. The tables shown in his article are for the data registers, D, but are usable also for address registers, A.

Incidentally I am not sure why 'D1-D2-D3', 'D2-D5-D6', 'D0,D2-D5-D6' and 'D0-D1-D6' appear in Norman's decoding table.

I must admit that I had not thought of using a table for the decoding of a register list. Instead I used code, which is certainly shorter but perhaps a wee bit convoluted.

I thought it might be of interest for others to see what this code is, so I append it, prefaced by a few remarks.

A Few Remarks

1. My code deals with floating point registers as well as integer unit registers. I thus have to distinguish between "D", "A" and "FP". Obviously this complicated the code, but only slightly.
2. I don't think the code as it stands, even with the comments beside the instructions, is sufficient for instant or easy comprehension. (I had to look at the code several times before I realised what it was doing and how it did it!) Hence the remaining comments indicate the method used, so that it should be easier to understand.
3. The array of bits in D0.B has been set at entry to the routine so that the most significant represents register 0, and the least significant, register 7.
4. Register D4 is used to contain the number of the next register to be looked for, from 7 back to 0. As we go through the bits in D0, D4 is decreased.
5. Register D3 contains one of "D", "A" and "FP". This is used to print the register type when needed. This is done by the subroutine at RGL_S.
6. The output is sent to the address to which A0 points at the outset of the routine. At the end of the routine A0 has been advanced to point to the next free byte in the print buffer.
7. At RGL_2, near the top of the routine, we are looking either for the first register in the list, or the first register after the end of a contiguous set. When we find a register we print it (eg "A3" or whatever). If there are more registers in D0 we start examining them at RGL_5. At this stage we stop the examination either when we run out of registers (D4 = 0) or when we reach the end of a contiguous set. In either case we print "-" followed by the register and a slash. If there are more registers to be examined we go back to RGL_2, otherwise we end the process by discarding the last slash.

RWAP Services NEWS!

New Websites!

We are proud to present our new websites! Starting September, we have separated our business in three separate websites.

You can find them at:

<<http://www.rwapadventures.com/>> (Adventure Games)
<<http://www.rwapservices.co.uk/>> (General Site) and
<<http://www.rwapsoftware.co.uk/>> (Software)

Our old address: <<http://hometown.aol.co.uk/RWAPSoftware/>> is of course still functioning but will be deprecated in the future.

New Products!

QWORD 1.0 NOW WITH DIGITAL SOUND ON QPC2!

The wait is now over! Q-Word version 1 is finally available!

Platforms:

QPC/QXL, Q40/Q60, Aurora (with SGC)

Prices:

All versions without P-Word £20.00p
All versions with P-Word £30.00p

Notes:

Q-Word **DOES NOT** require SMSQ/E with GD2 support -OR- SMSQ/E at all on the Aurora or Qx0 machines. It works on the highest colour depth everywhere regardless of Operating System.

The Aurora version is available on either HD or ED disk. For the latter add £1.00 to the price. ED version is uncompressed and can be run directly from the floppy. All other Floppy versions are compressed. QPC/QXL version comes on CD. Non CD versions DO NOW support digital sound on QPC2

Quantum Leap ED Drives

After many years of unavailability, here they are again! These are high quality Mitsubishi constructed/IBM badged drives with full warranty.

Unlike previously sold ED drives, these do not require FLP_JIGGLE and have no problems formatting 720K disks. However for the latter a switch is included with the cable.

ED Bare unit (no cable)	£ 29.99p
ED Boxed unit (complete with cable/ PSU)	£ 98.99p
Single unit Cable (with switch)	£ 4.99p
Dual unit Cable (with two switches)	£ 5.99p

(More options available, contact us for details. Also available mass quantities of ED and DD disks!)



for Windows

For QLers that run Windows or with incompatible hardware for Talent Games, we now have re-released these adventures so that they can run on your Windows-equipped PC. No Emulator, floppies, microdrive backups etc. required, just a one-click install! Of course the full QL line is still available! (See side column)

Talent Games for Windows ea. £ 10.00p
(Each Game includes a runtime installation of QLAY-2 by Jimmy Montesinos)

RWAP Services

35 Chantry Croft, Kinsley, Pontefract, West Yorkshire, WF9 5JH, United Kingdom Tel: +44 1977 610509 (From the UK dial: 01977 610509)
Website: <http://www.rwapservices.co.uk>

We accept:



(For PayPal please add 4% to the total price. Cheques in £ sterling made payable to R. Mellor)

Old Favourites!

Utilities

Sidewriter v1.08	£ 10.00p
Landscape Printing (EPSON printers)	
ImageD v1.03	£ 10.00p
3D object generator	
Q-Help v1.06	£ 10.00p
Superbasic On-Screen help system	
Q-Index v1.05	£ 5.00p
Keyword-to-topic finder	
ProForma ESC/P2 Drivers v1.04 for ProWeSs	£ 8.00p
Printer Driver	

Applications

Flashback SE v2.03 (upgrade only)	£ 5.00p
Database	
QL Cash Trader v3.7	£ 5.00p
Accounting/Finance	
QL Payroll v3.5	£ 5.00p
Accounting/Finance	
QL Genealogist v3.26	£ 20.00p
Genealogy	
Genealogy for Windows	£ 50.00p
QL Genealogist to Windows version upgrade	£ 25.00p
QL Cosmos v2.04	£ 5.00p
Planetarium	
Q-Route v2.00	£ 25.00p
Route Finding	
Upgrade from v1.xx	£ 5.00p
Britain map v1.11	£ 2.00p
BIG Britain map (needs 2Mb) v2.03	£ 5.00p
Various Britain Area maps (ask for details)	ea. £ 2.00p
Ireland map v1.00	£ 5.00p
Belgium map v1.01	£ 2.00p
Catalonia map v1.03	£ 2.00p
P-Word UK English Dictionary (500.000 words!)	£ 15.00p
Dictionary	

Leisure

Return to Eden v3.08	£ 10.00p
Adventure	
Nemesis MkII v2.03	£ 8.00p
Adventure	
The Prawn v2.01	£ 8.00p
Adventure	
Horroray v3.1	£ 8.00p
Adventure	
West v2.00	£ 5.00p
Adventure	
The Lost Kingdom of Zkul v2.01	£ 5.00p
Adventure	
All 6 games above	£ 25.00p
D-Day MkII v3.04	£ 10.00p
Strategy/War Simulation	
Grey Wolf v1.08	£ 8.00p
Graphical Submarine Simulation	
War in the East MkII v1.24 (upgrade only)	£ 5.00p
Strategy/War Simulation	
Open Golf v5.20	£ 8.00p
Sports Simulation	
QuizMaster II v2.07	£ 5.00p
Quiz	
Stone Raider II v2.00	£ 5.00p
Arcade Game	
Hoverzone v1.2	£ 5.00p
Arcade Game	
Deathstrike v1.5	£ 5.00p
Arcade Game	
Flightdeck v1.0	£ 10.00p
Flight Simulation	
All 6 games above (Open Golf, QuizMaster II, Stone Raider II, Hoverzone, Deathstrike and Flightdeck)	£ 28.00p

Notes on Software requirements

The following programs have a minimum SGC card requirement: P-Word, Qword, Big Britain MAP for Q-Route

My Code For Disassembling Register Lists

```
;
; At entry
;
;     D3.B = 'A' or 'D' or D3.W = 'FP'
;     D0.B = array of bits
;     A0 points to the first free byte in the print buffer
;
; At exit
;
;     The print buffer now contains the register list and
;     A0 points to the first free byte after that
;
RGL     TST.B     D0
        BNE.S     RGL_1
        RTS                ; Exit immediately if no list

RGL_1   MOVEQ    #7,D4      ; Initial value of D4

RGL_2   BTST     D4,D0      ; Is there a register? . .
        BEQ.S    RGL_3      ; . . no - go to next (or exit)

;
; Here we have found a first register, either on its own
; or at the start of a contiguous list
;
        BSR      RGL_S      ; Set A, D or FP in string
        SWAP    D0          ; Use top of D0 for the . .
        MOVE.B  #55,D0      ; . . register . .
        SUB.B   D4,D0       ; . . number
        MOVE.B  D0,(A0)+    ; Put it in the string
        SWAP    D0          ; Restore D0.B
        TST.B   D4          ; Finished? . .
        BNE.S   RGL_4       ; . . no
        MOVE.B  #"/", (A0)+ ; Set separator . .
        BRA.S   RGL_3       ; . . and exit

RGL_4   SUBQ.W  #1,D4       ; Go to next
        BTST    D4,D0       ; Register? . .
        BNE.S   RGL_5       ; . . yes
        MOVE.B  #"/", (A0)+ ; Set separator . .
        BRA.S   RGL_3       ; . . and go to next

;
; We test here for contiguous registers
;
RGL_5   TST.B   D4          ; Finished? . .
        BEQ.S   RGL_6       ; . . yes
RGL_7   SUBQ.W  #1,D4       ; Step to next
        BTST    D4,D0       ; Register? . .
        BNE.S   RGL_6       ; . . yes

;
; End of contiguous registers
;
        MOVE.B  #"-", (A0)+ ; Put "-" in string
        BSR     RGL_S       ; Set A, D or FP in string
        SWAP    D0          ; Use top of D0 for the . .
        MOVE.B  #54,D0      ; . . register . .
        SUB.B   D4,D0       ; . . number
        MOVE.B  D0,(A0)+    ; Put it in the string
        SWAP    D0          ; Restore D0.B
```

```

        MOVE.B    #"/", (A0)+ ; Set slash
        BRA.S    RGL_3      ; Go to next (or exit)
;
; We come here from the end of contiguous registers
;
RGL_6   TST.B    D4          ; Ended? . .
        BNE.S    RGL_7      ; . . no
        MOVE.B   #"_", (A0)+ ; Set "-" in string
        BSR     RGL_S      ; Put A, D or FP in string
        MOVE.W   #"7/", (A0)+ ; Put "7/" in string
RGL_3   TST.B    D4          ; Ended? . .
        BNE.S    RGL_8      ; . . no
        SUBQ.L   #1, A0     ; Throw away the last byte entered . .
        RTS
;
; Go to next register after end of contiguous list
;
RGL_8   SUBQ.W   #1, D4     ; Go to the next
        BRA.S    RGL_2
;
; Subroutine to enter A, D or FP in the string
; D3.B = "A" or "D" or D3.W = "FP"
;
RGL_S   CMPI.B   #"P", D3
        BNE     RGL_S1
        MOVE.W   D3, (A0)+
        RTS
RGL_S1  MOVE.B   D3, (A0)+
        RTS

```

Machine Code Tally Sort

Stephen Poole

One of the good features of the original QL was the beginner's guide tutorial, especially the self test, set at the end of each chapter, which ensured that the learner achieved a good grasp of the entire range of typical programming problems. So, in the summer of 1984, I got to chapters 13 and 16 dealing with sorting and wrote a short routine to sort whole numbers. When it was finished, I named it 'Tally Sort', stored it on micro-drive and forgot about it.

In march 1997, Dilwyn Jones wrote an article in QL Today presenting a certain number of sorting routines, including timings of the fast Quicksort and even faster Pigeon sort. Out of curiosity, I dug out the 'Tally Sort', timed it and was surprised to note that it easily outpaced the pigeon sort. So Dilwyn published tally-sort in the may 1997 QL Today, but in a letter to me he said he thought it was an April Fool's joke, as he received it on April 1st, and thought the name was a play on his address, which is of course Tal-Y-Bont!! But some readers tried it and assured him that it worked..

The listing is very short because the principle is very simple: Set up an array with the number of cells equal to the highest number to be sorted. Add a tally ('1') to the cell corresponding to every number to be sorted. Finally, starting from cell '0', look at every successive cell, and retain as many times the number as there are tallies, and repeat the process until the last cell has been inspected. Thus from low to high all occurrences of a number are found and are thereby sorted. The program is fast because it only requires two scans of the array... and spends most of its time printing the results, (if so required). On my JS SuperGoldCard with SMSQ/E, an integer% version sorts 32768 random numbers in 8 seconds. Of course in 1984 there were no go-faster add-ons for the QL, so the routine was much slower, and as I don't wish to disturb my SGC I can't re-do the original timings. But when I got Turbo, that gave a good boost, which has subsequently been eaten up by the accelerations of SGC and SMSQ/E.

Then, early last year, I began to consider machine-code programming and began buying books on the subject and using GWASS etc., but unfortunately none of them were for absolute isolated beginners like myself, so I wrote to Bruno Coativy who kindly offered me help with my

first Assembler project ... the Tally sort. To be perfectly fair, Bruno did most of the work, with me trying to follow his efforts step by step. Finally, Ian Pizer gave me '68000 Assembly Language Programming' from Osborne_McGraw-Hill, (with his Aurora system, offered in QI Today!), an excellent book which I can highly recommend as it contains ... a Tutorial! It only remained for me to translate Bruno's Assembler listing into English and convert his _bas and _bin files to work with a Basic loader. Bruno commented that it was not easy to get significant figures timing the machine coded sort, so the final program calls the code 100 times, effectively sorting 3,276,800 random

numbers. On my system the routine sorts over 200,000 numbers per second, and will print out accurate results for your own computer set-ups. I have not yet timed it on my new AMD 2800+ system as I do not have QPC2 yet, but it should be considerably faster still...

This article is only a demonstration of the 'Tally' method, so you will have to modify the assembler listing if you wish to use it seriously. That is, unless you get a copy of Bruno's Superbasic Extension 'TALLY BOOT' from the QLCF or Quanta Libraries. Indeed this could be adapted to sort more or larger numbers, but as Bruno says, the basic point of this article is to be pedagogic.

```
; Tally-sort Assembler Listing, by Bruno Coativy, v11july2003
; Source assembled using Assembler Workbench from Quanta.
```

```
maxint    EQU        32768-1        values go from 0 to 32767
Z         EQU        0              = ADRESZ - ADRESZ
X         EQU        4              = ADRESX - ADRESZ
Y         EQU        8              = ADRESY - ADRESZ
```

```
; This is the entry point
LEA       ADRESZ,A0        A0 is the Z buffer address
LEA       LBLZ,A1         A1 is the Z buffer
MOVE.L   A1,Z(A0)         .. save the address
ADD.L    #65536,A1        A1 is the X buffer
MOVE.L   A1,X(A0)         .. save the address
ADD.L    #65536,A1        A1 is the Y buffer
MOVE.L   A1,Y(A0)         .. save the address
MOVE.L   Z(A0),A1        A1 points to unsorted data
MOVE.L   X(A0),A2        A2 is the X array
MOVE.W   #maxint,D1      zeroise ...
```

```
RAZX
CLR.W    (A2)+           ... all 32768 ...
DBF      D1,RAZX         ... of array X ...
MOVE.L   X(A0),A2        A2 is the X array
MOVE.W   #maxint,D1      D1=32767
```

```
SCAN1
MOVE.W   (A1)+,D2        Take the current value ...
EXT.L    D2              ... which is the index ...
LSL.L    #1,D2           ... to increment ...
ADD.W    #1,0(A2,D2.1)   ... its array X frequency ..
DBF      D1,SCAN1       ... for each element.
MOVE.L   Y(A0),A3        A3 points to sorted data
MOVE.W   #maxint,D1      D1=32767
```

```
SCAN2
MOVE.W   #maxint,D2      D2=32767
SUB.W    D1,D2           The sort is done ...
MOVE.W   D2,D4           ... upwards.
EXT.L    D2              The current counter value D1 ...
LSL.L    #1,D2           is to calculate the index D2 ...
MOVE.W   0(A2,D2.L),D3   for the frequency count D4.
BRA.S    GETVALUE
```

```
THISVALUE
MOVE.W   D4,(A3)+        D3 times the value of D4 ...
```

GETVALUE

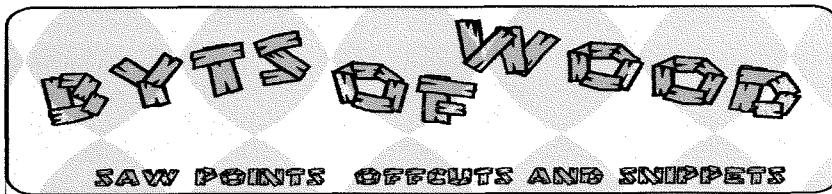
DBF	D3,THISVALUE	sort in the Y buffer ...
DBF	D1, SCAN2	... until it is 0.
MOVEQ	#0,D0	No errors.
RTS		Return to SuperBasic.
ADRESZ	DS.L 1	buffer Z address
ADRESX	DS.L 1	buffer X address
ADRESY	DS.L 1	buffer Y address

LBLZ

```
; No need to reserve 3 * 64k, just get it using ALCHP
; LBLZ DS.W 32768 Buffer of numbers to sort
; LBLX DS.W 32768 Tally Frequency table
; LBLY DS.W 32768 Sorted number buffer
```

END

```
110 REMark Tally_job_bas by B.Coativy v11july2003 and S.Poole v24mar2004
120 CLEAR: main: STOP: REMark LRUN under QDOS, or EXEC in SMSQ\E.
130 :
140 DEFine PROCedure load_array(base,max%)
150 LOCAL i%: PRINT ' POKEing 32767 random numbers into RAM using SuperBASIC'
160 FOR i%=0 TO max%
170     n%=RND(0 TO max%): POKE_W base+2*(i%),n%: AT 2,2: PRINT i%
180 END FOR i%
190 END DEFine
200 :
210 DEFine PROCedure showem
220     LOCAL i,n,pk: PRINT\\: n=base+length+32768*4
230     FOR i=0 TO 65536-2 STEP 2: pk=PEEK_W(n+i): PRINT pk,
240 END DEFine
250 :
260 DEFine PROCedure LOADem(base)
270 LOCAL i,rd: RESTORE 450
280 FOR i=base TO base+length-1: READ rd: POKE i,rd
290 END DEFine
300 :
310 DEFine PROCedure main
320 LOCAL length,base,max%,d1,loop,D2,i$
330 OPEN#1,con_16: WINDOW 512,256,0,0: CLS
340 length=124: base=ALCHP(length+3*64*1024): max%=32767
350 LOADem base: load_array base+length,max%
360 d1=DATE: IF d1=DATE: GO TO 360: ELSE d1=DATE
370 PRINT\\' Sorting 100 times over = 3,276,700 numbers...'
380 PRINT\\' Then comes the sorted list...'
390 FOR loop=1 TO 100: CALL base
400 D2=DATE-d1: showem: BEEP 1234,5
410 PRINT\ D2!'secs'\3276800'/D2!'sorts per sec'
420 i$=INKEY$(#1,-1): CLS: WINDOW 256,206,256,0: RECHP base
430 END DEFine
440 :
450 DATA 65, 250, 0, 110, 67, 250, 0, 118, 33, 73, 0, 0, 211, 252, 0, 1, 0, 0, 33
460 DATA 73, 0, 4, 211, 252, 0, 1, 0, 0, 33, 73, 0, 8, 34, 104, 0, 0, 36, 104, 0, 4
470 DATA 50, 60, 127, 255, 66, 90, 81, 201, 255, 252, 36, 104, 0, 4, 50, 60, 127
480 DATA 255, 52, 25, 72, 194, 227, 138, 82, 114, 40, 0, 81, 201, 255, 244, 38
490 DATA 104, 0, 8, 50, 60, 127, 255, 52, 60, 127, 255, 148, 65, 56, 2, 72, 194,
227
500 DATA 138, 54, 50,40, 0, 96, 2, 54, 196, 81, 203, 255, 252, 81, 201, 255, 230
510 DATA 112, 0, 78, 117, 0, 7, 120, 60, 0, 8, 120, 60, 0, 9, 120, 60
520 ::
```

To an extent we are all prejudiced for and against certain things. As human beings none of us can offer the fabled 'unbiased opinion'. This is evident in the discussion that I had on the user group list with John Sadler about operating systems. I apologise if some of you have read this on the list and now see some of it rehashed here but it is something that fascinated me. I also apologise that it mentions Windows so much but, since we are all computer users of one kind or another it seems appropriate that we should consider, and learn from, other systems.

A Little History

When most of us started using computers they could not do much more than display a bit of text on the screen. Anything complex had to be typed in laboriously and then, if you were lucky, saved onto tape. This led to a lot of tightness of the code produced with people trying to squeeze as much functionality out of as small a piece of code as was possible. As time passed we all saw an exponential increase in the amount of computing power available to us and the amount of space available to store it all on. This has made today's computers very powerful with large fast storage capacity and facilities undreamed of in the eighties when the QL was born.

These advances have often been made at the expense of user comprehensibility. Whereas, in the dawn of the home computer, many of the people using it could understand how

it achieved what it did, people today have little comprehension of what is 'under the bonnet'. Some of this is due to the extreme size and complexity of the code and some is due to the fact that today's computers are sold to people who don't really want to know how it works. This is not a criticism of the general public, after all how many people understand the principle behind the common biro? They just want to use it.

Here Comes the Bogeyman

No-one likes a success and the success of the Windows operating system has attracted more than it's fair share of vitriol over the years. To an extent some of this was warranted some of the bricks hurled through the corporate 'windows' were well deserved. Microsoft has never been the model of caring corporations and some of the tricks it has used have been very underhand if the reports were anything to be believed. Some of its software has also been far from perfect. Many people see this as being something they should be up in arms about to a far greater degree than the transgressions deserve and complain that they are 'being forced to use substandard software'

Well, as I said on the list no-one forces you to use anything at all. Most Windows software is fairly stable now and my laptop running XP rarely crashes. In fact these days I have had more crashes running SMSQ/E! This however is not what I

wanted to talk about. The part of the discussion that fascinated me and led to this preamble is the evolution of the Operating System.

Pointers to the Origin

In the older days of the QL most people could not write a boot file and had no idea the system was multitasking because they would just stick the disk in and reboot whenever they wanted to use a new program. We have come a long way from all of that and most commercial software these days is not sub standard at all.

The O/S barely existed at all to start with. If we ignore the older systems which read punched cards the true O/S came about with the early UNIX command line structure. Most file handling and graphics output was handled by a separate routine run independently. Memory was small, expensive, and very slow and hard drives were rare. As memory became cheaper and more abundant and CPUs increased in their ability and speed people began to think of things that they wanted to have in the O/S. This is about the time that the computer moved out of the labs and onto the desktop. This is an important move because it took the computer out of the hands of pure scientists and put it into the reach of people like you and me.

Now, if you are crunching a long sequence of numbers in an environment which is slow and lacks memory you want all of your resources to concentrate on the task at hand and not wander off to play CDs or watch movies. This is natural and understandable. When the computer escaped from the labs it began to be used by amateurs, people who wanted to play with its possibilities.

They were not calculating Pi to ten thousand decimal places, some of them, in fact, wanted to play 'Pong'. These were also the people who could not recall all of the commands available to the command line and they wanted to load files easier. They started nagging the programmers for a better user interface and, being programmers, they liked the challenge. OK, the O/S moved forward a few steps.

CPUs got faster, memory got cheaper and the amount of free capacity for machines not running serious scientific processes increased. Programmers were hooked into the idea that maybe they could add a few more useful things too and began to add stuff to the O/S. The concept of a functioning O/S without file handling, sound and rudimentary graphics became unthinkable so they were added as standard.

The same thing is true of the hardware scene. As many of us recall, the original motherboards came with nothing on board at all. Most of them needed an extra card to perform that basic drive I/O. These days the onboard I/O is a standard feature for most boards and many come with on board graphics, sound, LAN, and some even have RAID and a built in modem. This is usually because the chips used to provide one function have enough spare capacity to perform all the other tasks. Nasta has noted this growing trend while designing the new SuperIde card so much so that he has added the LAN port to his design even though there is no specific software support for it. The hope is that, once the port is available someone may write the drivers.

When is an Iguana not an Iguana?

As Darwin noted, things evolve. What started out as a couple of pegs to hang your clothes on became a whole new wardrobe with some very fancy togs in it. This is where the purists diverge and I have not argument with their concept that the system should be completely under the control of the user. What I would say is that, given that most users can barely use some of the supplied software anyway, giving them the options to control their systems is bound to lead to complete disaster. Systems like Windows diverged from the more 'pure' computer systems that many of us have grown up with because it decided to try to provide a 'complete' package. You want a sound player? Here it is built in. You want to play movies? Here is a media centre. You want to browse the Internet? Here is your browser built in and ready to use when you switch on. OK, this may be flawed and have bugs but bugs are a way of life in the computer business. Now, having this all built in may make the machine a little slower but so are some of the users and those that learn fast soon learn to turn off stuff they do not want. More to the point here is the fact that I know people who have a computer and have no word processor or spreadsheet on it. It is an internet terminal, a games machine and a DVD player and nothing else!

Now some of the policies bolted onto the media player may be controversial such as the Digital Rights Agreement. This gets up many people's noses but only because they want to pirate films and music and this software is designed

to stop peer to peer file sharing. It won't, of course, because whatever can be made can be broken but that is another story. I have already launched into my 'free software' argument in prior columns so I won't rehash it here except to say that if the author wants it free to distribute then that is fine and if he wants to protect his investment that is also fine.

So Windows is now a different kind of Iguana than those found basking around Lake LINUX or any of the other habitats for wild computer O/S's. I fail to understand why people make such a fuss about that, especially now that it is the boss of IKEA who is the worlds richest man. Maybe that is the next whipping horse.

Good Systems

Before I drag this back to QL terms I would like to mention a system here that I always considered to be one of the best. This was the system provided on the PSION Series 3. This had most of the most useful utilities readily available on the ROM. Word Processing, File Handling Diary, Calendar, Time and Spreadsheet functions were right there at the push of a button if you wanted any more than this you could get 3rd party software to do it. Not as easy to type on given its small screen and keyboard (although I did write a large chunk of one of these columns on mine while flying back to the UK from Eindhoven in Peter Fox's plane).

My PC boots up pretty fast and runs pretty well because it does not load the Windoze programs at start up as many people believe. No computer does. Even the components hard coded into Windoze are not running, they are called and

closed down as needed. In some ways the QL system can be more wasteful if you consider the size of the application against the size of the operating window (i.e. RAM/hard drive - I speak about hardware QLs and not things like QPC2). If you take the QL as an example (and that is what we should be discussing but this thread is interesting in that it shows the misconceptions people are under.) If I start an Aurora QL and LRESPR SMSQ/E I have a basic O/S with colour support and pointer support. No applications are installed or running. If I want to run my own invoicing program I will have to LRESPR menu_rext ProWesS, and QLIB_run. Now I have some more memory being used, my BOOT time is longer and the system is slightly slower. No actual programs are running yet but some program code is loaded and available should I call it. The same is true for many other programs which have their own extensions or need to be loaded resident.

If I want to make all of those programs available at the click of a mouse or by hitting a HOTKEY I must issue some other commands in the boot file which will remain in the system and add another cycle of slowness to it. That is the downside. The upside is that I no longer have to type EX..... to get something to work so the time added in the boot up is shaved off over the period you use the computer. Magnify this several fold and you have the modern day computer. It loads more extensions, more dlls and masses more drivers because it will need them for the programs and devices attached but it also has a much faster CPU and has much more capacity than a QL so it is pretty much equal in these terms.

Now I enjoy messing around with my QL and I would not be on the QL Users forum, attending QL shows and spending hours writing for the magazine if I did not. It is a simple system and one man can understand it but it is also limited. No QL word processor is the equal of Wordpad and, on the whole, the PC is over-brimming with good, well written software I wish we had a fraction of it.

We dwell in a different world to that of the average Windoze user in that we use our computer for all manner of things. Many of these are obscure uses for which there are no mainstream applications written and many are just for the sheer joy of playing around with code. It does not matter if you have a number of different computer systems for a number of uses. Never forget it is a tool and not a religion. If many of the people who spend a lot of time carping on about how bad M\$ is used that time to develop software for the QDOS/SMSQ scene then we would have a bit more to play with!

Golden Clives

You may have seen, in the last issue, a mention of the 'Golden Clives' awards. We decided in the 20th anniversary of the QL's birth that we would try to honour some of those who have stuck with the system over the years and who have contributed a lot to the fact that it is still being used. We would really like our readers and any other people they are associated with in their local user groups to send in votes for those who might get the awards. Over the twenty years that the QL has been in existence there have been various prominent members of the community who have pro-

duced some great software and hardware for us to use and now is your chance to give them a vote of thanks for their hard work. Send your votes either on paper or electronically to Jochen Merz and the awards will be announced at the QL 2004 show in Eindhoven.

Notes From The American Show

I am writing this having just returned from the US show. This year the show was held in Florida which meant that we had some good weather and sizzling temperatures to deal with. Although the show was not as well attended as it could have been we did have some interesting discussions and presentations. Tony spent a lot of his time there repairing various bits of QL hardware and I spent a lot of time breaking QDT. I am very impressed with the way that QDT has developed since it was first shown to me and I am pleased to say that Jim hopes to have it available for release at the Eindhoven 2004 show. It is now on my desktop here and being used properly for the first time.

I was even more impressed when Jim, with a little help from Marcel was able to create an icon from my QBranch logo so I have that as a folder on my desktop now. I can foresee some opposition to this though from the very purists I mentioned above. These are the people who do not believe in using a mouse or in having a graphical interface at all. In my view, though, this is vital development in the way that the computer is used. Having programs and files available at the click of a mouse makes the whole process of producing and using data on a computer

so much easier once the skills involved in manipulating a mouse have been gained.

Some of the bugs I found with the system were supposed to have been fixed by the weekend after the show but Jim came down with a damaged elbow the day after the show and had to go into hospital for a small operation a week later. He was temporarily reduced to one handed typing. At the time he suggested that this was down to a spider bite and, since we spent the day before at the Universal Studios park and had a couple of shots at the Spiderman 3D ride, I began to wonder if the next show would see him hanging from the ceiling from a web and rushing off to save the world halfway through the meeting. I gather he is well on the way to recovery now and I look forward to a full implementation of QDT soon.

Another Damaged Medium

Talking about QL programmers who have been laid out I would like to extend my best wishes to Bernd Reinhardt who has a skiing accident earlier this year and now has more metal in his legs than the average family saloon car. The accident put paid to his plans for visiting the Hove show this year since he is expected to be unable to travel for several months. I hope you will all join with me in wishing him a speedy recovery from his injuries. I also hope that he is putting all this sitting down time to good use in developing QL programs (tee hee).

Beach Bums Basking in Brainwaves

Although, as I said above, we had to endure high tempera-

tures in Florida and were forced to sit on the beach sipping Margheritas we did manage to get some interesting discussions on QL projects going. In between swimming and shopping that is.

One of the good things about this annual trip is that Jochen, Marcel and I spend a week together and, during that time, we discuss a lot of the current QL issues and try to make plans for upcoming projects.

We had some quite good ideas for new facilities whilst doing this and I hope that some of these ideas will make it into the new versions of the software that will be produced over the coming months. One of the ideas was for a 'history device' in the QMenu files system. This will allow you to have a list of recently opened/saved files when you call the 'Files' menu. Another concept was to add a 'variables' listing to the Superbasic side of QD so you can see at a glance what variables you have used. Very useful when writing large programs.

Given our recent musings on the user group list, printing was high on the agenda. I spent a lot of time discussing this in the last column so I will not dwell on it too deeply here but we did resolve to try to make some progress in settling the software issue during the next year. Indeed, I think this is vital to the continued survival of the QDOS/SMSQ system. The hardware issue will be something which is harder to resolve but that is also something which needs some serious thought.

USBs In Your Bonnet

QPC 2 has less of an issue with this than native Hardware systems because it can use the USB ports to talk to prin-

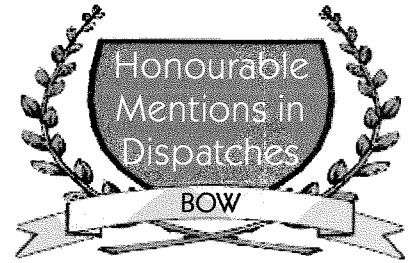
ters but even these need the appropriate software support. Rich Mellor contacted me about using ProWesS to talk to modern printers and I agree to an extent that a ProWesS style driver would be one way to go but it does still need a driver for each printer and, as such, a programmer to write that driver. Although the ProWesS system is based upon purely graphical objects it still talks to the printer via a driver and that driver is still printer specific.

Nasta also talked to the user list to say that providing a hardware USB port was not an insurmountable problem but again the software to run it would be tricky. The problem with USB, as I see it, is that it is designed to do a multitude of things and, within that remit, it has to make intelligent decisions on what is attached to it and how it is handling the devices. One thing that often confuses people who are more QL oriented is how to allocate the ports. I have had many people with new PCs which have four or more USB ports on them asking how they tell the computer to print to which port. The answer, of course, is that the port itself does all that.

Having said all this there is no reason why we should have to have such a complicated setup for QDOS/SMSQ systems. If you accept that, for the moment at least, we will not be plugging anything other than a printer into the port a simplified version of the driver should be able to handle the throughput. Maybe this is something for Tony Firshman to look into. There must be a way to produce a box which will take standard nine pin serial output from Aurora or Q40 boards and convert it to USB for a printer to use. Such a box could even be used on a standard QL with the appropriate

cables. Maybe the board could even be mounted on a PC case back plate and just use a header onto the Aurora/Q40. The problem would then be 'just' a driver/printer software one. Of course this is a big 'just' because it does involve the way that the printer expects feedback from the computer. Many printers these days use a system whereby the amount of ink used is calculated and displayed on the com-

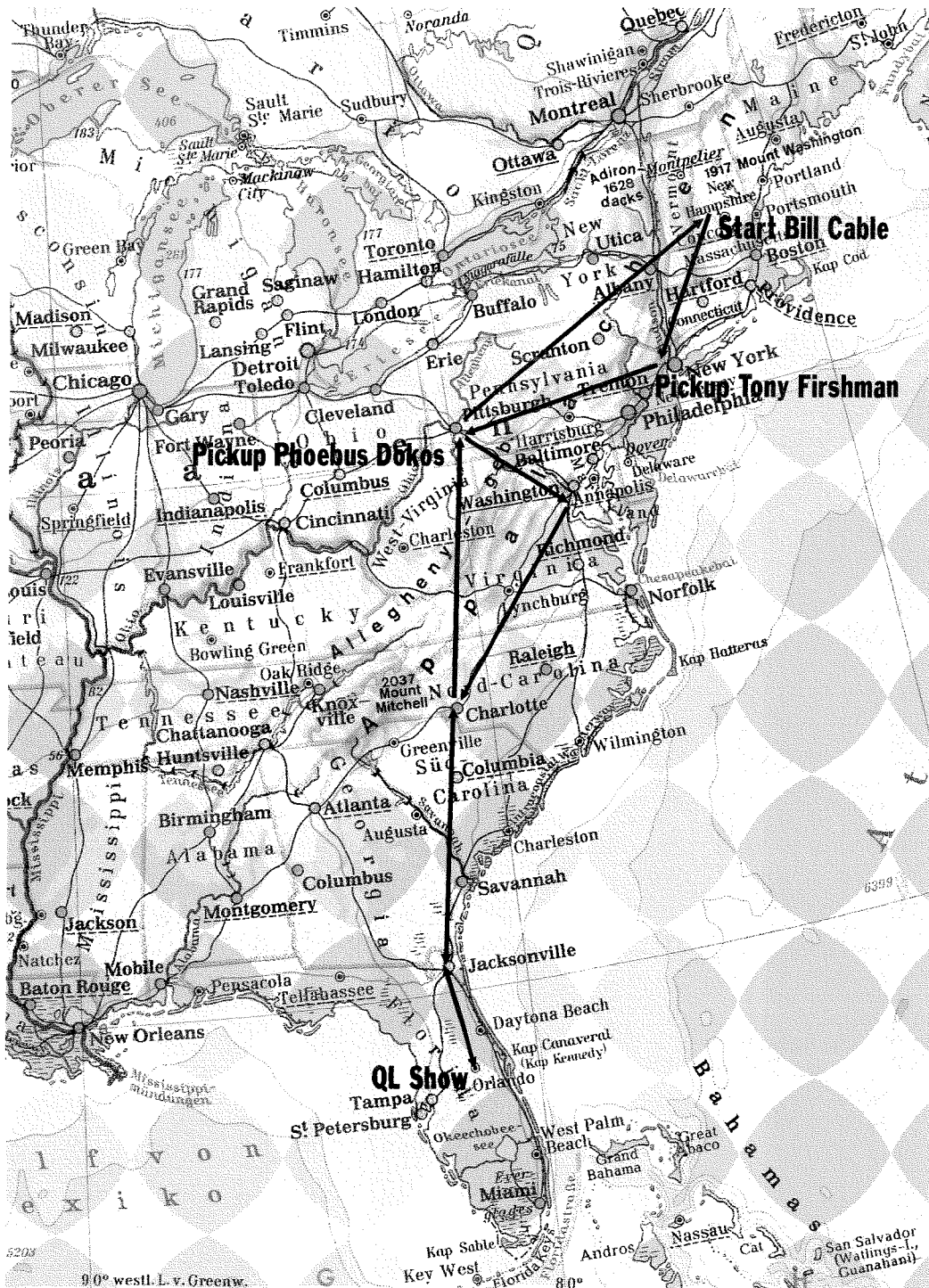
puter screen. This information is then used to tell the printer if it can print or if a tank needs replacing. Many modern printers have a multitank system with separate ink tanks for each colour. This does mean that it is harder to work out which ones are low without the software display. There is always another hurdle to overcome but some of these do get jumped.



Honourable Mentions in Despatches

This month's honour goes, not for a piece of code or hardware but for a feat of endurance.

I took the easy route to the US QL show. I flew direct from London to Miami and then had a short car ride to the Holiday Inn which is directly on the beach. Given Tony Firshman's ability with maps he decided to fly to New York and drive down to Florida with Bill Cable. Somehow or other someone had the idea that they should pick up Phoebus 'on the way'. They drove for 36 hours with only fuel and food stops and arrived at the hotel in Orlando early on Friday evening. Straight after the show was over we went out for a meal and then they set off to repeat the procedure in reverse. So the matchsticks under the eyelids award goes to Bill, Tony and Phoebus.



QL2004

EINDHOVEN

16th October

LIGHT CITY CHALLENGE!

The QL2004 team invite you to celebrate the QL's 20th birthday at an international workshop to be held in Eindhoven, the "Light City" of the Netherlands.

We shall be looking back on 20 years of the QL, but also looking forward. Our Light City Challenge is to take the QL into its 21st year and beyond.

VENUE

Pleincollege St. Joris, Roostenlaan 296, Eindhoven, Netherlands.

Saturday 16th October from 10.00 to 17.00.

The Pleincollege St. Joris is one of the oldest and most popular QL workshop venues with a large hall for the main meeting and smaller rooms for demonstrations and lectures. The hall is well provided with electric sockets, but please bring your extension leads etc. For demonstrations we can make use of projectors, and it is helpful if your system can be connected to these.

There are no cafes or restaurants in the immediate vicinity of the show venue, but the show hosts, the Dutch user group Sin-QL-Air, will provide light refreshments at cost price throughout the day.

ACTIVITIES

Most of the main traders have indicated they will be present at the show, and there will be ample room for you to set up your own system. In addition we are planning several workshops and a QL Forum.

A QL Internet Connection Workshop will explore the latest progress in developing emailing from your QL, and a Colour Workshop will look at tools and techniques for using the new colours in your own programs.

We also hope to have some Q60 related activities and news of expansion cards for native hardware. We would also like to see some progress made in improving QL printer compatibility.

To end the formal proceedings, there will be a QL Forum with a panel of "experts". We are still inviting nominations for this panel, and would like some questions for the panel in advance.

BY CAR

Travel to Eindhoven on the Venlo - Antwerp Motorway (A67 - E34) and leave at Knooppunt Leenderheide following the signs Centrum. You are now in the Leenderweg. At the first roundabout turn left into Floralaan West. Stay on this road to the first traffic lights where you should turn left. You are now in the Roostenlaan. Look for the signs to Pleincollege St. Joris, which is on the right hand side of the road. There is ample parking space on the school campus.

PUBLIC TRANSPORT

There are frequent trains to Eindhoven from all corners of the Netherlands.

For train times and prices go to www.ns.nl, click on the English link and use the snelplanner.

To catch a bus to the show venue, turn right on leaving the station platform and go to the bus station at the rear of the railway station. You need bus line 7 in the direction of Waalre, which is an half hourly service. Ask for the stop Roostenlaan/Floralaan. The journey is over 2 zones, (i.e. 3 strips of your strippenkaart) and takes just over 10 minutes. When you leave the bus, turn left, walk over the Floralaan and keep walking straight on. Pleincollege St. Joris is about 5 to 10 minutes from the bus stop on your righthand side.

If you prefer to use the taxi, then you should turn left when you leave the station platform and go to the front of the station. (Please note there is no longer a treintaxi service in Eindhoven.)

**More information on the
QL2004 website - see below.**

Geoff Wicks, 56 Peveril Crescent, West Hallam, Derbyshire DE7 6ND, U.K.

Tel: +44 (0)115 - 930 3713

email: gwicks@beeb.net

Web: <http://members.lycos.co.uk/geoffwicks/ql2004.htm>



The QL Show Agenda



QL Meeting - (NL) Eindhoven

Saturday, 12th of June, 10:00 to 16:00
Pleincollege St. Joris, Roostenlaan 296
Same venue as always!

Jochen Merz Software and Just Words will be there,
and QBranch and TF Services after a long time of abstinence too.
Last chance to meet the QL2004 team in person before the event itself!

QL Meeting - (D) Berchtesgaden

Saturday, 2nd of October, 10:00 to 17:00
Hotel Schwabenwirt, Königsseer Str. 1
Same venue as last year!

All details will follow in the next issue. We will try to keep it shorter, as
the general situation and description of the beautiful area of
Berchtesgaden has not changed, of course. Ideal to add a holiday!
The computer room in the hotel will be available again from Friday
afternoon to Sunday morning, and locked during the night.
A nice, social event with Dinner after the show.

OCTOBER,

16TH

**DETAILS
REVERSE
ON
SIDE!**

QL 2004